Understanding Endogenous Data Drift in Adaptive Models with Recourse-Seeking Users

Bo-Yi Liu*[1], Zhi-Xuan Liu*[1], Kuan Lun Chen[1], Shih-Yu Tsai[2], Jie Gao[3], Hao-Tsung Yang[1]

[1]National Central University, Taiwan
[2]National Yang Ming Chiao Tung University, Taiwan
[3]Rutgers University, U.S.
boyiliu1007@gmail.com, asd185622@gmail.com, milochenckl@gmail.com,
shih-yu.tsai@nycu.edu.tw, jg1555@rutgers.edu, haotsungyang@gmail.com

Abstract

Deep learning models are widely used in decision-making and recommendation systems, where they typically rely on the assumption of a static data distribution between training and deployment. However, real-world deployment environments often violate this assumption. Users who receive negative outcomes may adapt their features to meet model criteria, i.e., recourse action. These adaptive behaviors create shifts in the data distribution and when models are retrained on this shifted data, a feedback loop emerges: user behavior influences the model, and the updated model in turn reshapes future user behavior. Despite its importance, this bidirectional interaction between users and models has received limited attention. In this work, we develop a general framework to model user strategic behaviors and their interactions with decision-making systems under resource constraints and competitive dynamics. Both the theoretical and empirical analyses show that user recourse behavior tends to push logistic and MLP models toward increasingly higher decision standards, resulting in higher recourse costs and less reliable recourse actions over time. To mitigate these challenges, we propose two methods—Fair-top-k and Dynamic Continual Learning (DCL)—which significantly reduce recourse cost and improve model robustness. Our findings draw connections to economic theories, highlighting how algorithmic decision-making can unintentionally reinforce a higher standard and generate endogenous barriers to entry.

Code — https://github.com/asai-lab/Understanding-Endogenous-Data-Drift-in-Adaptive-Models-with-Recourse-Seeking-Users.git

1 Introduction

Deep learning has emerged as a fundamental tool in decision-making and recommendation systems, typically structured into two main phases: *training* and *prediction* (Zhang et al. 2019). In a standard binary classification scenario, during the training phase, the model captures patterns from historical data. In the prediction phase, the trained model is deployed within a real-time system—such as YouTube's recommendation engine or e-commerce platforms—to drive decisions or recommendations (Qin et al. 2020; Kirdemir et al. 2021; Zhou 2020; Shankar et al. 2017).

This paradigm relies on the assumption that the data distribution remains unchanged post-deployment. However, in practice, deployed systems often influence the data they consume. For example, users receiving unfavorable outcomes (e.g., rejection) may engage in *recourse actions*: modifying their attributes to better align with the model's criteria, typically at minimal cost (Karimi et al. 2022; O'Brien and Kim 2021; Nguyen, Bui, and Nguyen 2023; Poyiadzi et al. 2020; Yadav, Hase, and Bansal 2022; Venkatasubramanian and Alfano 2020). A prominent example of this phenomenon is the proliferation of websites and articles that offer strategies to "beat" an algorithm, teaching users how to exploit their mechanics on recommendation systems such as Google, YouTube, Facebook, and so on (MacDonald 2023; Klug et al. 2021).

However, recourse actions tend to be strategically narrow rather than diverse, for two main reasons. First, available "slots"—such as job positions, loan approvals, or top-ranked recommendation slots—are inherently limited and highly competitive, forcing users to tailor their efforts toward aligning with the system's criteria rather than exploring varied improvement paths. (Herlocker et al. 2004; Hennig-Thurau, Marchand, and Marx 2012). Second, users tend to focus on modifying low-cost, high-impact features. For example, optimizing for engagement metrics like clicks and likes because such changes offer relatively high returns with minimal effort, rather than investing in more substantive improvements (Chen, Wang, and Liu 2023; Estornell et al. 2023). Although these behaviors may not be explicitly dishonest or rule-breaking, they can nevertheless generate unintended and potentially harmful distortions in the data. Indeed, studies have shown that social media recommendation systems can amplify polarization and emotionally charged content, further undermining the assumption of a static data distribution post-deployment (Chitra and Musco 2020; Chen, Wang, and Liu 2023).

While some research addresses the influence of user recourse behavior and data drift in deployed models (Hardt and Mendler-Dünner 2025; Altmeyer et al. 2023), few works explore how such recourse influences model evolution when

*These authors contributed equally.

the system adapts to new data. Our assumption is that it would amplify the deviated direction and becomes a vicious circle, creating a feedback loop: behaviors adapt to the model, the model retrains on these behaviors, thus reinforcing the skew. As an example in video recommendation, Yuval Noah Harari (Harari 2024) explains that videos with extremist content tend to drive higher user engagement. Thus, a recommendation system designed to maximize user retention may encourage and promote such content in the end. Empirical evidence supports this claim. A systematic review shows that 21 out of 23 recent studies implicated YouTube's recommendation system in promoting problematic content pathways (Yesilada and Lewandowsky 2022). These findings underscore the bidirectional influence between user behavior and model updates.

In this work, we investigate the interaction between system deployment and user recourse behavior, and how this dynamic affects both model updates and user features over time—referred to as model shift and data shift (Hardt et al. 2015). This interaction is inherently bidirectional: the deployed model influences users' recourse actions, leading to changes in the data distribution (data drift), and in turn, this data drift causes the model to update (model shift), completing a feedback loop. To the best of our knowledge, this is the first work to discuss the interaction in the long run. Two prior studies are most relevant to our setting. One focuses on the resource competition between recourse users when calculating the recourse actions (Fonseca et al. 2023). The other explores recourse algorithms and conditions that make the model shift but does not investigate the bidirectional long-term effects and properties triggered by the interactions between the system and users (Altmeyer et al. 2023).

To address this gap, we develop a framework that explicitly models the full interaction loop: *model deployment* $\rightarrow$ *user response (i.e., recourse actions)* $\rightarrow$ *model update*. A key observation in this setting is that the system operates under limited resources and cannot accommodate all users, even when many improve their features. As a result, the system must update its decision criteria based on the new, competitive data distribution. Our study focuses on two central questions:

1. *Under limited resources, how do recourse users influence the system and drive long-term model updates?*
2. *When model updates are driven by recourse users, how do these changes affect future users and their recourse behavior over time?*

Our Contributions: We examine the iterative dynamics between user recourse behavior and model updates. Our key findings are summarized as follows:

1. We provide a theoretical analysis showing that logistic models tend to evolve toward a higher decision standard over time. This trend is further supported by simulation results across a range of settings.
2. We show that a higher decision standard leads to increased recourse costs and results in less reliable, more failure-prone recourse actions.
3. To address these challenges, we propose two strategies—Fair-top-k and Dynamic Continual Learning (DCL)—which effectively mitigate the identified issues and enhance model robustness.

Notably, the first two findings can be related to multiple economic principles. For instance, competitive environments often push systems toward producing higher-quality or higher-standard outcomes—analogous to the model adopting a stricter decision boundary in our setting (Bikker and Haaf 2002; Ezrachi and Stucke 2015). Furthermore, such dynamics can generate endogenous barriers to entry, making it more difficult for new or less advantaged users to succeed (McAfee, Mialon, and Williams 2004; Pehrsson 2009). This directly relates to our second finding: as the model standard rises, recourse actions become increasingly costly and are more likely to fail over time.

Lastly, our proposed methods show strong empirical results. Fair-top-k and DCL significantly alleviate the higher standard issue, improving performance by approximately 50% across all rounds. Recourse costs are reduced by around 70%, and balanced accuracy remains stable, reaching approximately 90% in the long run—demonstrating improved robustness compared with other methods.

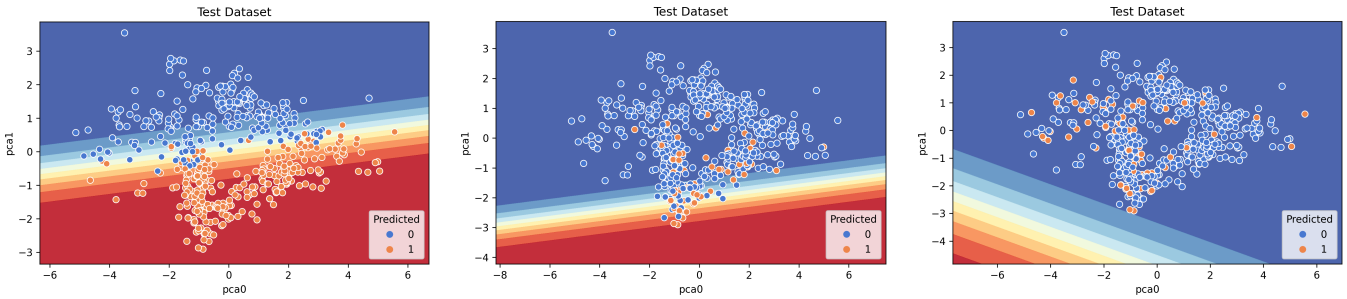
Experiment: Figure 1 shows the experimental results with a binary logistic model. The experimental setup is described in Section 6. Simulations show that the model classifies accepted and rejected data points around half and half during the first few rounds. However, as the number of rounds increases, the decision boundary shifts to a higher standard and rejects majority of users who do not execute recourse actions.

2 Related Work

Algorithmic recourse offers interpretable and actionable guidance to help individuals alter model outcomes, enhancing transparency and accountability in machine learning (Gunning et al. 2019; Arrieta et al. 2020; Guidotti et al. 2018). In high-stakes domains, such as finance (Barocas, Hardt, and Narayanan 2017), healthcare (Bastani, Kim, and Bastani 2018; Bertossi 2020; Liang et al. 2014), and education (Fiok et al. 2022; Khosravi et al. 2022), recourse mechanisms help individuals understand the reasoning behind model predictions and identify feasible improvement steps.

Most algorithmic recourse methods are designed for one-hop decisions, using gradient-based optimization to identify minimal input changes that flip the model decision (Shao and Kersting 2022). FACE uses shortest-path methods with density-based metrics (Poyiadzi et al. 2020), while GDPR emphasizes user-centric explanations that support legal and ethical accountability (Wachter, Mittelstadt, and Russell 2017).

In a real-world setting, users adapt their behavior in response to model suggestions, influencing future model inputs. Over time, such feedback-driven adaptation can lead to distributional changes. These observations motivate the integration of continual learning into recourse-aware systems, enabling models to adjust dynamically to users' changing behaviors.



(a) Original distribution (b) Updated at 30-th. round (c) Updated at 70-th. round

Figure 1: The experiment of model evolution with algorithmic recourse, where the model is updated with the top-k labeling. These figures show the test dataset, which is sampled from the original distribution and remains unchanged across all rounds.

2.1 Data Drift and Continual Learning

Even without the bi-directional effect on the model and recourse users, it is naturally to have concept shift or data drift when deploying the model over time. There are several work address this issue for the algorithms generating recourse actions. Gao et al. proposed a generative algorithm to mitigate social segregation risks in large-scale algorithmic recourse (Gao and Lakkaraju 2023). Gupta et al. addressed fairness under data shift by equalizing actionable recourse across demographic groups (Gupta et al. 2019). Other work such as Rawal et al. (Rawal, Kamar, and Lakkaraju 2021), Upadhyay et al. (Upadhyay, Joshi, and Lakkaraju 2021a), and De Toni et al. (De Toni et al. 2025) address the reliability of the recourse actions under the concept drifts over time.

From the deployed model side, Continual learning (CL) is particularly relevant for adapting to different distributional dynamics because it allows models to learn from a stream of tasks without catastrophic forgetting (Lopez-Paz and Ranzato 2017a; Zenke, Poole, and Ganguli 2017; Shi et al. 2024). Common strategies include regularization-based approaches (Kirkpatrick et al. 2017), replay-based methods that store or generate past samples for rehearsal (Rolnick et al. 2019; Rebuffi et al. 2017), and dynamic architecture approaches that expand the model structure to accommodate new tasks (Rusu et al. 2016; Yoon et al. 2018).

However, these approaches assume that task sequences are externally defined and overlook evolving data distributions. In contrast, our setting is behavior-driven: data adapt to model outputs, leading to feedback-driven distributional changes, which blurs the boundary between continual learning and data shift.

2.2 Model Deployment with Recourse

Beyond distributional concerns, recent studies highlight the need for robust algorithmic recourse under model shifts and real-world deployment dynamics. Proposed approaches focus on ensuring recourse coverage, stability under distributional changes, and resilience to feedback loops that may degrade model performance over time (Bui et al. 2025; Upadhyay, Joshi, and Lakkaraju 2021b; Rawal et al. 2023). Other work focus on the algorithms of generating recourse action, which tends to incentives users to perform truthful improvement rather than deceiving the system (Chen, Wang, and Liu 2023; Estornell et al. 2023). Unlike designing new recourse algorithms, some work observe and discuss the influence the of recourse users to the deployed model such as increasing classification errors (Fokkema, Garreau, and van Erven 2024) or affecting reliability and fairness over time (Bell, Fonseca, and Stoyanovich 2024). Altmeyer et al. examined evolving models and data shifts in the recourse setting and proposed mitigation strategies (Altmeyer et al. 2023).

To our knowledge, none of these works investigates the long-term bidirectional effects and properties triggered by the interactions between the system and users. Our proposed research builds on these simulation-based studies and extends them to explore their long-term impacts on both users and systems. We design strategies to mitigate these effects, and foster positive feedback loops benefiting both sides.

3 Framework

We discuss the decision-making scenario with a set of users and a binary classification model in multiple rounds. Each user is a data point with d features and the model is a score function between 0 and 1 (0 means "rejected" and 1 means "accepted"). In round t, a dataset $\mathcal{D}^t$ with size N is sampled from a distribution P_{data}, which is a mixed distribution between $\mathcal{D}^{t-1}$ and $\mathcal{D}^0$ (i.e., the original distribution). Then, $\mathcal{D}^t$ is modified to $\mathcal{D}'^t$ in which some of the rejected users in $\mathcal{D}^t$ perform recourse actions to modify their features and improve their scores in h^t, hoping to flip the results. The system receives the (responded) data set $\mathcal{D}'^t$ and labeled it based on the scores of h^t and the constrained resource k. Lastly, the model is updated to h^{t+1} based on $\mathcal{D}'^t$ with the labels. The procedure of round t is shown in the following.

1. **Sample the dataset:**
$$\mathcal{D}^t = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}, \quad \text{with } x^{(i)} \sim P_{\text{data}}.$$
2. **Randomly select rejected users and perform recourse action:**
Randomly select a subset of rejected users $\mathcal{S} \subset \mathcal{D}^t$ and $\forall x \in \mathcal{S}, h^t(x) < 0.5$. Modify the features of each user x using recourse function r. The modified set $\mathcal{S}'$ is:
$$\mathcal{S}' = \{r(x) \mid x \in S\}.$$

The updated dataset $\mathcal{D'}^t$ is:

$$\mathcal{D'}^t = (\mathcal{D}^t \setminus \mathcal{S}) \cup \mathcal{S}'.$$

3. **Determine new labels for $\mathcal{D'}^t$ with resource k:** For each user $x \in \mathcal{D'}^t$, the label is determined by the labeling function f, which converts the predicted score $h^t(x)$ into a value in $[0, 1]$ with the hard constraint that the number of accepted samples in $\mathcal{D'}^t$ is at most k. That is, for each $x \in \mathcal{D'}^t$, $f \circ h^t(x) \in [0, 1]$ and

$$\sum_{x \in \mathcal{D'}^t} \mathbb{1}[f \circ h^t(x) = 1] \leq k$$

4. **Update the model:** Use the modified dataset $\mathcal{D'}^t$ and the new binary labels $\{y^{(i)}\}$, where $y^{(i)} = f \circ h^t(x^{(i)})$, to update the model by minimizing the loss function $\mathcal{L}$:

$$h^{t+1} = \arg\min_h \sum_{x^{(i)} \in \mathcal{D'}^t} \mathcal{L}(h(x^{(i)}), y^{(i)}).$$

This iterative process allows us to explore how recourse actions influence the system dynamics, user behavior, and the evolution of the recommendation model over time. We are specifically interested in Step 2, 3, 4, which we sequentially denoted by *User Response phase*, *Labeling phase*, and *Model Update phase*.

3.1 User Response Phase

The user response function r on model h^t can be understood via the recourse action analysis (Verma et al. 2020; Upadhyay, Joshi, and Lakkaraju 2021b). The recourse action x' for a user with features x is determined by solving the following optimization problem:

$$x' = \arg\min_{x'} c(x, x'),$$
$$\text{s.t.} \quad h^t(x') = 1,$$

where $c(x, x')$ represents the cost function associated with the action. However, this equation is generally challenging to solve due to the presence of a hard constraint. To address this, most approaches reformulate the problem by Lagrangian relaxation:

$$x' = \arg\min_{x'} \ell(h^t(x'), 1) + \lambda c(x, x'), \tag{1}$$

where $\ell : [0, 1] \times [0, 1] \to \mathbb{R}$ is a differentiable loss function (e.g., binary cross-entropy), ensuring that the gap between $h^t(x')$ and the favorable outcome 1 is minimized. The parameter $\lambda > 0$ acts as a trade-off between minimizing the loss and the cost of the recourse action. In this framework, the quality of recourse can be controlled by replacing the favorable outcome 1 with some constant p less than 1. Usually, p should be larger than 0.5 to ensure the recourse feature x' can be at least accepted in h^t. For example, one typical setting is to consider the L_2 distance with constant coefficients (i.e., c is a vector with all positive values).

$$x' = \arg\min_{x'} \ell(h^t(x'), 1) + \lambda \sum c_i (x'_i - x_i)^2 + c_0 \tag{2}$$

3.2 Labeling Phase

Given the constrained resource k and predicted score $h(x)$, a classical policy is to apply the top-k function (Fonseca et al. 2023). In this case, the function f sets the largest k values of $h(x)$ to 1 and the rest to 0:

$$y = f \circ h(x) = \begin{cases} 1 & \text{if } h(x) \text{ is among the top } k \text{ values} \\ 0 & \text{otherwise} \end{cases}$$

3.3 Model Update Phase

Here we consider two settings, a typical setting and continual learning setting. Both settings use ADAM (Kingma 2014) as the optimizer, where the typical setting uses the cross-entropy loss. The continual learning uses its specific loss function mentioned in Section 5.2.

4 Analysis on Logistic Model Evolution

We investigate the influence of recourse users on model evolution. Specifically, we aim to understand how recourse actions contribute to the internal momentum that shapes the direction of model updates over time. Our analysis focuses on the logistic regression model, which serves as a fundamental building block for understanding more complex deep learning architectures.

Definition 1 (Resource saturation)**.** *Follow the framework in Section 3, the system is said to reach* ***resource saturation*** *at round t if the accepted users in D' is exactly k.*

In the subsequent analysis, we assume that the system remains fully saturated at all time steps.

Definition 2 (Higher standard)**.** *Given two binary classifiers h, h' and a data distribution $\mathcal{D}$, we say that model h has a* ***higher standard*** *than model h' if and only if:*

$$\mathbb{E}_{x \sim \mathcal{D}}[h(x)] < \mathbb{E}_{x \sim \mathcal{D}}[h'(x)].$$

Theorem 1 (Higher Standard Provides Higher Accuracy)**.** *Let D, h be a dataset and the logistic model respectively at round t. For the responded dataset D', there exists a model h' that has a higher standard than h and achieves higher accuracy on D' if and only if there exists at least one recourse user, newly labeled as positive (i.e., class 1), whose score ranks within the top-k in D'.*

The proof is provided in the Appendix. In summary, if a recourse user enters the top-k ranked scores in D', the score threshold corresponding to the k-th user in D' is higher than that in the original dataset D. As a result, the decision boundary of h can be shifted to this higher threshold, thereby rejecting all users in D' with scores below the kth ranked user. This effectively increases the model's standard (i.e., makes it stricter) and results in improved classification accuracy over D'.

While Theorem 1 illustrates that pursuing higher accuracy may lead to a more strict decision standard, accuracy is not the typical optimization objective in most learning settings. Therefore, we turn our attention to cross-entropy loss (CE loss), the standard objective in logistic regression, and examine how recourse behavior influences model evolution under this loss.

Lemma 1 (Recourse action)**.** *Given a logistic model h and a L_2 cost function c (Equation 2), the recourse action taken by a user on feature i is proportional to $\frac{w_i}{c_i}$, where w_i and c_i denote the weights associated with feature i in the model and the cost function, respectively.*

Proof. Denote $h(x) = 1/(1 + e^{-(wx+b)})$. Referring to Equation 2, the cost of action a on recourse function r is in the following.

$$r(a) = -(\log h(x + a)) + \lambda \sum c_i a_i^2 + c_0.$$

Since r is convex, we can calculate the optimal action a via the partial derivative is zero at any dimension i. That is,

$$\frac{\partial r}{\partial a_i} = -w_i(1 - h(x + a)) + 2a_i c_i \lambda = 0.$$

This implies

$$a_i = \frac{1 - h(x + a)}{2\lambda} \cdot \frac{w_i}{c_i}$$

□

Lemma 1 illustrates that recourse actions are biased in proportion to $\frac{w_i}{c_i}$. Based on this insight, our first observation is that model updates driven by failed recourse users tend to result in a new model h' with a higher standard under the cross-entropy loss. This is formally stated in Proposition 1.

Proposition 1 (Failed Recourse Actions Drive Higher Standards)**.** *Under the cross-entropy loss, let D be a dataset and h be the logistic model trained on D. If a user x performs a recourse action resulting in $\hat{x}$, but is still assigned label 0 in the updated dataset D', then the resulting model h' will have a higher standard than h.*

Proof. Since $\mathcal{L}_{CE}(h(x), y)$ is convex, the partial gradient of it is zero among all dimensions over D. That is, $\forall i \in [1, d]$

$$\sum_{x \in D} \frac{\partial \mathcal{L}_{CE}(h(x), y)}{\partial w_i} = 0,$$

which gives

$$\sum_{x \in D^+} x_i(h(x) - 1) + \sum_{x \in D^-} x_i(h(x)) = 0, \tag{3}$$

where D^+, D^- denote the set of users with label 1 and 0, respectively. Assume $w_i, x_i > 0$. In Equation 3, the first term contributes the negative gradient and the second term contributes the positive one, denoted by $\mathcal{H}^+_{D,h}, \mathcal{H}^-_{D,h}$ respectively. One can think the value of w_i is changed in order to balance the two terms, i.e., sum to zero.

Denote the recourse user before and after the recourse action as x and $\hat{x}$. Notice that both x and $\hat{x}$ belong to D^-, D'^- but $\hat{x}$ has a higher score and feature value in dimension i (via Lemma 1). Thus, the value of $\mathcal{H}^-_{D',h}$ is higher. Now, consider the situation that the updated model h' is found via gradient descent starting from the parameters of h. The direction of changes in w_i is to balance $\mathcal{H}^+_{D',h}, \mathcal{H}^-_{D',h}$. Since x_i is unchangeable, the adjustment only affects the output score of x. That is, after h' fits to D', we have

$$\sum_{x \in D^+} x_i(h(x) - 1) > \sum_{x \in D'^+} x_i(h'(x) - 1)$$

and

$$\sum_{x \in D^-} x_i(h(x)) < \sum_{x \in D'^-} x_i(h'(x)).$$

The above inequalities show that the updated model h' provides lower score than h. □

On the other hand, if the resource is shortened, even if there is no recourse action, the updated model h' also leads to a higher standard.

Proposition 2 (Limiting Resource Drives Higher Standard)**.** *Under the cross-entropy loss, let D, h be the dataset and the logistic model fit to D. If a user x labeled as 1 in D but 0 in D' due to the limited resource, it drives the updated model h' into a higher standard.*

One can prove Proposition 2 by a similar analysis in Proposition 1, which we skip here.

However, the system becomes complicated when some recourse users replacing the non-recourse users via top-k policy. To see that, denote by u the user labeled as 1 in D and replaced by some recourse user $\hat{x}$ via top-k policy. With a similar analysis with partial derivative on i, we have

$$H^-_{D',h} = H^-_{D,h} - x_i h(x) + u_i h(u),$$

and

$$H^+_{D',h} = H^+_{D,h} + \hat{x_i}(h(\hat{x}) - 1) - u_i(h(u) - 1).$$

Reorganizing,

$$H^-_{D',h} + H^+_{D',h} = u_i - x_i h(x) - \hat{x_i} + \hat{x_i} h(\hat{x}).$$

Denote the above term by $F(x_i, \hat{x_i}, h(x), h(\hat{x}), u_i)$. The positivity of F subject to the following constraints.

$$\begin{aligned} \text{Subject to: } & 0 \le x_i < \hat{x_i} \\ & h(x) < h(\hat{x}) \\ & 0 \le h(x) < .5 \\ & .5 \le h(u) \le 1 \\ & .5 \le h(\hat{x}) \le 1 \end{aligned}$$

Due to the complexity of F, which involves 5 variables and product of variables, we explicit list some conditions that make F positive or negative.

$$F(x_i, \hat{x_i}, h(x), h(\hat{x}), u_i) \begin{cases} > 0, & u_i \ge (1 - \alpha)\hat{x_i} \\ > 0, & x_i < u_i < \hat{x_i} \text{ and } \frac{x_i}{\hat{x_i}} > \beta \\ < 0, & u_i \le x_i \text{ and } \frac{x_i}{\hat{x_i}} < \beta \\ < 0, & u_i < \frac{1}{2} x_i, \end{cases} \tag{4}$$

where $\alpha = h(\hat{x}) - h(x), \beta = \frac{1-h(\hat{x})}{1-h(x)}$.

Generally, we observe three key patterns. First, if the replaced user has a significantly higher feature value (i.e.,

$u_i > x_i$), the resulting model h' tends to adopt a higher standard; conversely, if the value is much lower (e.g., $u_i < \frac{1}{2}x_i$), the standard tends to drop. Second, the greater the score improvement from the recourse action, the more likely h' is to shift towards a higher standard. Third, when the change in the feature value x_i is disproportionately large compared to the resulting score change—indicated by a high β value—h' tends to adopt a higher standard. The second and third observations suggest that intense and biased recourse behavior can drive the model toward a more strict decision threshold.

5 Methods Toward Robust Model Evolution

Section 5.1 and Section 5.2 are our two proposed methods to address the issue of standard shifting.

5.1 Fair Top-k Policy

To address the issue of standard shifting, the intuition is to select a set of users that are "diverse" from each other. That is, if a set of points with high scores is similar to each other, we would decrease the chance of selecting all of them as accepted points. To do so, we use kernel density estimation (KDE) to assess the density of similar data points, which provides a measure of local crowdedness. Next, we generate a biased weight vector v by combining each kernel density score inverse with $\kappa\hat{y}_i$, where κ is a dimensionless weighting factor that balances the contribution of $\hat{y}_i$. Using this weighted distribution, we randomly select k data points from those where $f(\hat{y}_i) = 1$. The remaining unselected data points are assigned $y_i = 0$, indicating rejection.

$$v_i = KDE^{-1}(x_i) + \kappa\hat{y}_i \tag{5}$$

However, after the labeling phase, the accepted and rejected data become mixed due to random selection, which can negatively impact model training. To mitigate this issue, we remove some negative data points from the training set if they satisfy $h(x_i) > 0.5$.

5.2 Continual Learning and DCL

To prevent model from drastic shifting, we use continual learning as a method to memorize past distributions. Specifically, we adapt Synaptic Intelligence (SI) (Zenke, Poole, and Ganguli 2017), a classical continual learning method to achieve our goal. The original SI is a regularization-based continual learning method, with loss regularization defined as follows:

$$\tilde{\mathcal{L}}_t = \mathcal{L}_t + \tau \sum_k \Omega_k^t (\tilde{\theta}_k - \theta_k)^2 \tag{6}$$

$$\Omega_k^t = \sum_{u<t} \frac{\omega_k^u}{(\Delta_k^u)^2 + \epsilon} \tag{7}$$

In equation (6) , τ is a dimensionless scaling factor that regulates the contribution of the previous task's weight. θ_k represents each individual parameter of the current parameter set and $\tilde{\theta}_k = \theta_k(t-1)$. Ω_k^t is the per-parameter regularization strength. In equation (7) , ω_k^u is the per-parameter loss in each task while $\Delta_k^u = \theta_k(u) - \theta_k(u-1)$. ϵ is a small value to prevent zero-division.

In our scenario, we are not interested in the loss among all past distributions but only a few past-rounds. To focus on short-term tasks rather than all tasks before task t, we modify the function Ω_k^t by introducing a learning range r and a dimensionless weight constant w_u. The weight w_u follows a cubic distribution, assigning greater importance to tasks closer to t.

$$\Omega_k^t = \sum_{u=t-r}^{t-1} w_u \frac{\omega_k^u}{(\Delta_k^u)^2 + \epsilon} \tag{8}$$

The previous continual learning setting set constant value τ among all rounds, which is not flexible when facing different data distributions. To address this, we introduce **Dynamic Continual Learning (DCL)** by modifying τ to

$$\tau_t = \frac{\tau}{JSD_{t-1}} \tag{9}$$

Here we use the Jensen-Shannon Divergence (JSD) $\in [0, 1]$ (Lin 1991) as an evaluation metric to measure the distance between the positive and negative data distributions. It allows the strength of past tasks to be adjusted based on the previous round's level of chaos. Hence, if the last task is nearly collapsed, JSD_{t-1} decreases, leading to an increase in τ_t. This prevents the model from aggressively learning new patterns that could introduce further instability into the model.

6 Experiments and Observations

The experiments aim to simulate a virtual scenario where the decision model that incorporates a recourse function over the long term, starting with training on a fixed distribution of data points. Our experimental setup follows the framework introduced in Section 3. We use classical two-layers MLP and logistic regression as the decision models on three different datasets. The resource constraint k is set to $\frac{N}{2}$. The labeling phase includes Top-k and our proposed method: Fair-top-k. In the model update phase, we use ADAM with Binary Cross Entropy (BCE) loss, continual learning loss, and dynamic continual learning (DCL) loss we described in Section 5.2. Both The constant τ used is 10^{-5} in DCL and typical continual learning. The hyperparameter κ for Fair-top-k is set as 10^{-4}.

We use both synthetic and two real-world data to simulate our virtual environment. The synthetic dataset is generated in $\mathbb{R}^{20}$, with 17 dimensions that are all actionable (mutable). *UCI defaultCredit* dataset (Yeh and Lien 2009) is related to customers' default payments and has 23 features with 19 of them are actionable. In *Credit* dataset we referred the work from Ustun et al. (Ustun, Spangher, and Liu 2019) and set 11 actionable features.

During the recourse phase, Equation 1 is used to generate the recourse action. The cost function is calculated among all mutable features. Denote $x =< x_1, \cdots, x_d >$ and $x' =< x'_1, \cdots, x'_d >$

$$c(x', x) = \sum_{i, i \in mutable} c_i \left(x'_i - x_i\right)^2 + c_0.$$

We set c_0 to 0 so there is no cost if no action is taken. In real-world datasets, the weights $\{c_i\}$ are set to reflect the

real-world scenarios. In the synthetic dataset, we try different weight distributions including uniform, inverse gamma distribution, and logarithm distribution.

6.1 Metrics

We use several metrics to evaluate the model stability, model robustness, and recourse fairness of the interaction between system and users.

Short-Term Balanced Accuracy We adapt the concept of Average Accuracy (AA) (Chaudhry et al. 2018; Lopez-Paz and Ranzato 2017b) and modify it to the evaluation focuses on recent performance. Let $a_{t,j} \in [0, 1]$ represent the classification accuracy of the j-th round task on the t-th round model.

$$AA_t = \frac{1}{t}\sum_{j=1}^{t} a_{t,j} \tag{10}$$

Given that data will become imbalanced over rounds, negatively impacting prediction accuracy, we replace $a_{t,j}$ with balanced accuracy $b_{t,j}$ (Brodersen et al. 2010). Balanced accuracy is calculated as the average of recall and specificity, making it a more suitable metric for imbalanced datasets. Here, the balanced accuracy $b_{t,j}$ is defined as:

$$b_{t,j} = \frac{1}{2}(\frac{TP_{t,j}}{TP_{t,j} + FN_{t,j}} + \frac{TN_{t,j}}{TN_{t,j} + FP_{t,j}}) \tag{11}$$

To capture the short-term focus, we introduce r to the equation, defining the number of past rounds considered in the calculation.

$$STBA_t = \sum_{j=t-r}^{t-1} b_{t,j} \tag{12}$$

Additionally, we exclude the current round from the calculation to ensure an unbiased evaluation of the model's true performance, as the model is trained on the t-th task.

Higher Standard The details of this metric are explained in Definition 2. Here, $\mathcal{D}$ represents the test dataset, and we track the value of the expected output logit before the sigmoid layer: $\mathbb{E}_{x\sim\mathcal{D}}[h'(x)]$. Notably, due to the sigmoid function's asymptotic nature, its output is not ideal for observation (values become very close in each round). Therefore, we analyze $h'(x)$, the model output logit prior to the sigmoid transformation. Furthermore, since the sigmoid function is monotonically increasing, this change does not affect the inequality relationships among higher standard values.

Average Recourse Cost For each recourse user, their recourse cost is $\sum_i c_i(x'_i - x_i)^2 + c_0$, as described in Equation 2. We set c_0 to 0 because there is no cost if no action was taken. This metric calculates the average recourse cost of each recourse user.

Fail to Recourse (FTR) Users care about the effectiveness of their recourse actions. While ensuring stability, system also needs to maintain fairness for recoursed users. Hence, we introduce Fail to Recourse (FTR) to quantify the overall effectiveness of recourse action, regarding the system. It calculates the proportion of recoursed data in round t that fail to be classified as positive in $t + 1$ round. While Fonseca et al. (Fonseca et al. 2023) proposed recourse reliability (RR) as the proportion of recoursed data in round t that classified as positive in $t+1$ round, we do it the opposite way. R_t denotes the subset of all recoursed data and N_{t+1} denotes the subset of classification result which is negative after $t + 1$ round of training.

$$FTR_t = \frac{|R_t \cap N_{t+1}|}{|R_t|} \tag{13}$$

Test-Acceptance Rate To quantify the increasing decision boundary, we introduce the Test-Acceptance Rate (TAR) as an indicator to observe the model's classification standards. Specifically, we examine the ratio of labels 1 and 0 at t-th round in the test dataset, which reflects the original distribution of the test data, to understand how the model's classification criteria change. Here T^1 denotes the data with label 1 and T^0 denotes the data with label 0 in the test data.

$$TAR_t = \frac{|{T^1}_t|}{|{T^0}_t|} \tag{14}$$

6.2 Top-k Labeling with Static Learning

Figure 2 shows the simulation result in the static learning when the top-k policy is used in labeling phase and cross-entropy loss is set in model update phase. The x-axis is the number of rounds, and the y-axis is the value of the measured metric, which from left to right are Higher Standard, Test Acceptance rate, and Short-term Balanced Accuracy. Each row represents one of the datasets. We have three observations.

Higher Standard occurred. As illustrated in Figure 2, the higher standard metric shows that the model outputs lower score, leading to stricter acceptance criteria across the three datasets, as evidenced by the decreasing test acceptance rates. The test acceptance rates across three datasets are all close to zero after a few rounds, where the value is generally greater than 0.5 in the initial distribution. The highest value is around 0.25 after a few rounds, shown from MLP model in synthetic dataset. This indicates that there are less than 20% samples that will be accepted after several rounds if they are from the initial distribution.

High recourse cost and a high failure rate are observed in MLP models. Figure 3a presents the average recourse cost for both the Logistic and MLP models. The MLP model yields a higher recourse cost compared to the Logistic model. The MLP model is more flexible to the new data distribution, so the model tends to shift more, therefore requiring users to exert more effort to satisfy its criteria, leading to higher recourse cost. Furthermore, the inherent flexibility of MLP model can also introduce instability issues. According to Figure 3b, the MLP model's Fail to Recourse is markedly higher than that of the logistic model. As indicated by the higher standard metric in Figure 2, the drastic changes in the MLP's output logits underscore the model instability and the difficulty in aligning recourse.

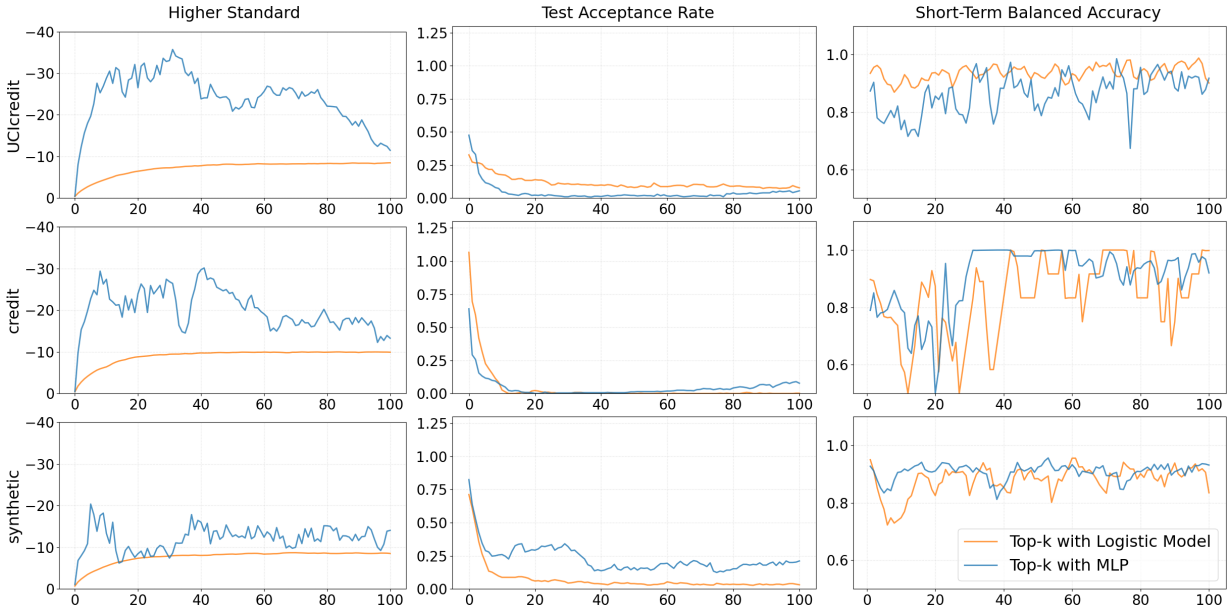
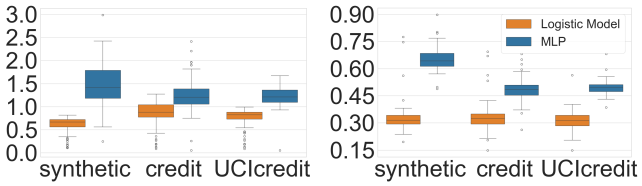


Figure 2: Higher Standard, Test Acceptance Rate, and STBA on Logistic Regression Model and MLP across three datasets.



(a) average recourse cost (b) FTR

Figure 3: (a) average recourse cost of Top-k method across three datasets. (b) Fail to Recourse (FTR) of Top-k method across three datasets.

Non-robust accuracy is observed in both models. According to Figure 2, the Short-Term Balanced Accuracy is notably low for the *Credit* dataset. This can be attributed to the data distribution of *Credit*, where positive samples in the *Credit* dataset are closer to the decision boundary. Consequently, the test acceptance rate for positive instances drops significantly, leading to only a few positive samples existing in the test dataset. Therefore, the balanced accuracy calculation is heavily penalized if the classification of these positive instances changes between rounds. This leads to a great fluctuation on all Short-Term Balanced Accuracies.

The above observations are also consistent across varying levels of recourse quality (from 0.7 to 1.0) and different ratios of recourse users (from 0.2 to 0.7). While other factors may also influence these patterns, due to space constraints, we provide a summary in Table 1 and include the full experimental details in the Appendix. Overall, increasing the recourse quality and the proportion of recourse users tends to raise the decision standard and reduce the robustness of the model's predictions.

Setting \ Situation	Non-robust model	Recourse cost	Recourse fail rate	Higher standard
Higher recourse quality	↑	↑	↓	↑
More recourse users	↑	↑	↑	↑
Larger k values	↓	–	–	↓
Inverse gamma cost weights	–	↓	–	–
Higher ratio of original distri.	↑	↓	↓	↑
From logistics to MLP	–	↑	↑	↑
From logistics to conti. learning	–	↓	–	–

Table 1: Effect of Different Settings on top-k Method.

6.3 Fair-top-k, Continual Learning, and DCL

This subsection compares our proposed methods (Fair-top-k and Dynamic continual learning) with the classical method (Top-k with typical update) and the typical continual learning(Top-k with continual learning update). The comparison shows similar trends across all three datasets, thus we report the results on the *Credit* dataset and put the rest of them in Appendix.

Figure 4 illustrates the comparison of five different strategies: Fair-top-k labeling with a typical update (orange), Fair-top-k labeling with a DCL update (red), Top-k labeling with a DCL update (green), Top-k labeling with a typical update (purple), and Top-k labeling with continual learning (blue).

Higher standard is eased among all settings. The

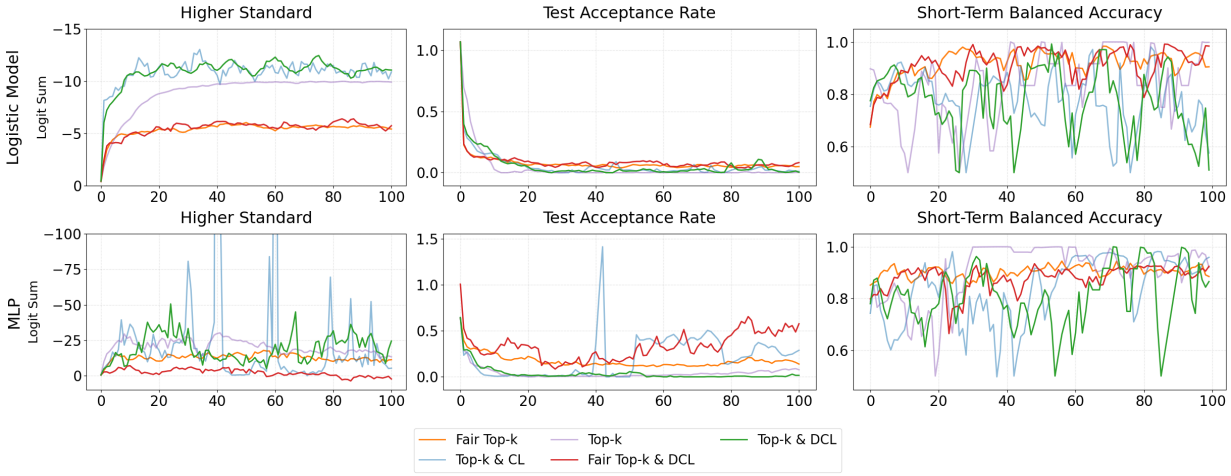


Figure 4: Higher Standard, Test Acceptance Rate, and STBA on Logistic Regression Model and MLP on *Credit* data. The outlier of Higher Standard on MLP with Top-k & CL method is -776.66, -132.24 at round 40, 41 and -360.13 at round 60.

Dataset	Methods	Average Recourse Cost		Fail to Recourse		Higher Standard	
		Logistic	MLP	Logistic	MLP	Logistic	MLP
Synthetic	Top-k	0.60 ± 0.02	1.46 ± 0.05	$\mathbf{0.32 \pm 0.01}$	0.65 ± 0.01	-7.64 ± 0.16	-12.05 ± 0.30
	Top-k & CL	0.25 ± 0.01	0.60 ± 0.03	0.33 ± 0.01	0.70 ± 0.01	-8.33 ± 0.10	-4.50 ± 0.50
	Top-k & DCL	0.38 ± 0.01	$\mathbf{0.55 \pm 0.02}$	0.33 ± 0.01	0.54 ± 0.01	-8.79 ± 0.11	-11.28 ± 0.43
	Fair-top-k	0.19 ± 0.01	1.05 ± 0.02	0.39 ± 0.01	$\mathbf{0.52 \pm 0.01}$	$\mathbf{-5.05 \pm 0.09}$	$\mathbf{0.47 \pm 0.11}$
	Fair-top-k & DCL	$\mathbf{0.15 \pm 0.00}$	0.61 ± 0.03	0.41 ± 0.01	0.53 ± 0.01	5.36 ± 0.10	$\mathbf{-0.46 \pm 0.46}$
Credit	Top-k	0.98 ± 0.02	1.24 ± 0.04	0.34 ± 0.01	0.46 ± 0.01	-8.98 ± 0.19	-20.18 ± 0.49
	Top-k & CL	0.58 ± 0.01	0.90 ± 0.05	$\mathbf{0.30 \pm 0.01}$	0.61 ± 0.02	-10.82 ± 0.13	-27.41 ± 8.51
	Top-k & DCL	0.56 ± 0.01	1.02 ± 0.05	$\mathbf{0.30 \pm 0.01}$	$\mathbf{0.43 \pm 0.01}$	-10.90 ± 0.15	-19.16 ± 0.96
	Fair-top-k	0.48 ± 0.01	1.33 ± 0.04	0.43 ± 0.01	0.45 ± 0.01	$\mathbf{-5.38 \pm 0.07}$	-12.74 ± 0.25
	Fair-top-k & DCL	$\mathbf{0.42 \pm 0.01}$	$\mathbf{0.26 \pm 0.02}$	0.43 ± 0.01	0.47 ± 0.01	$\mathbf{-5.51 \pm 0.08}$	$\mathbf{-2.05 \pm 0.24}$
UCIcredit	Top-k	0.75 ± 0.02	1.22 ± 0.03	$\mathbf{0.31 \pm 0.01}$	0.49 ± 0.01	-7.21 ± 0.18	-24.01 ± 0.62
	Top-k & CL	0.41 ± 0.01	0.53 ± 0.04	0.39 ± 0.01	0.65 ± 0.01	-7.08 ± 0.10	-2.91 ± 0.43
	Top-k & DCL	0.42 ± 0.02	0.54 ± 0.03	0.38 ± 0.01	0.61 ± 0.01	-7.16 ± 0.10	-7.10 ± 0.49
	Fair-top-k	0.23 ± 0.01	1.43 ± 0.04	0.45 ± 0.01	$\mathbf{0.46 \pm 0.01}$	$\mathbf{-5.15 \pm 0.08}$	-5.13 ± 0.23
	Fair-top-k & DCL	$\mathbf{0.18 \pm 0.00}$	$\mathbf{0.40 \pm 0.03}$	0.45 ± 0.01	0.58 ± 0.01	$\mathbf{-5.22 \pm 0.08}$	$\mathbf{1.10 \pm 0.19}$

Table 2: Average Recourse Cost, Fail to Recourse, and Higher Standard across three datasets on Logistic and MLP Models

Higher Standard metric across the five methods shows that both Fair-top-k solutions (orange and red lines) maintain values that are approximately 50% better than other methods, indicating a more stable and consistent standard throughout the rounds. In contrast, the methods without the fair-top-k policy (green, blue, and purple lines) exhibit a drastic decrease in both models. Furthermore, the fair-top-k solutions also achieve a higher Test Acceptance Rate, as implied by the stable Higher Standard. Overall, Fair-top-k labeling policy can ease the higher standard problem.

Recourse cost is significantly reduced and model accuracy is enhanced. In Table 2, Fair-top-k & DCL consistently show lower Average Recourse Cost than other methods. For example, the cost of our methods is generally 70% or lower than top-k. This metric exhibits a strong correlation with the increase in the standard; higher standards correspond to higher recourse costs. However, the cost-invalidity trade-off (Guo et al. 2023) persists, as observed in both Fail To Recourse and Average Recourse Cost. This means that a higher cost typically corresponds to a lower Fail To Recourse rate. Even within this trade-off, Fair-top-k & DCL still achieve a significant cost reduction, while at most incurring a 14% higher possibility of failing to recourse. On both models, Fair-top-k policies (with and without DCL) achieve a strong, stable and sustained short-term balanced accuracy of approximately 90% in the long term. For the MLP model, the top-k with typical update method achieves the best short-term balanced accuracy in later rounds on the MLP model. This is attributed to its model quickly shifting to a higher standard early on, as seen in the test acceptance rate drop-

Setting \ Situation	Non-robust Model	Recourse Cost	Recourse Fail Rate	Higher Standard
Higher recourse quality	↑	↓	↓	↑
More recourse users	↑	↑	↑	↑
Larger k values	↓	↓	↓	↓
Inverse gamma cost weights	–	↓	–	↓
Higher ratio of original distri.	↑	↓	–	↑
From logistics to MLP	↑	↑	↑	↓

Table 3: Effect of Different Settings on Fair-top-k Method.

ping to about 0 before 20 rounds, which then stabilized with minimal changes in later rounds.

Continual learning reduces the recourse cost. As shown in Table 2, methods employing a continual learning approach always achieve a lower cost compared to those using the same labeling policy without it. This advantage stems from continual learning's ability to retain knowledge from past models, ensuring the recourse actions remain stable and relevant over time.

Our proposed methods provides tradeoff solutions in MLP models.

The aforementioned three observations are also confirmed across other datasets, different recourse quality and ratio of recourse users. The summary of other factors is shown in Table 3 and the details are included in the appendix.

7 Discussion, Conclusion, and Future Work

In this work, we study model evolution in the presence of recourse users under limited system resources. Our analysis and experiments indicate that recourse behavior naturally pushes the model toward a higher decision standard, thereby increasing the difficulty of future recourse actions—reflected in higher recourse costs and failure rates. To address these challenges, we proposed the Fair-top-k strategy and a dynamic continual learning algorithm. Both methods significantly mitigate the problems identified and enhance model robustness, as evidenced by improved balanced accuracy over time in logistic models.

Nevertheless, this work represents only an initial step toward understanding the broader implications of recourse behavior in learning systems. There remain many open directions for further exploration, particularly in connection to fields such as strategic decision-making and social sciences. Below, we outline a few promising directions:

The prediction of the momentum in model evolution is still quite open. We showed that under certain conditions, model evolution tends to favor higher standards. However, it remains an open question whether one can theoretically characterize how recourse actions systematically influence the direction of model evolution. Specifically, what are the theoretical links between cost functions, recourse algorithms, and real-world constraints? On the other hand, the momentum in other configurations are still open, such as other models (e.g., random forests, complicated deep learning models, ... etc.), gaming framework, online & real-time system, extreme resource constraints (in our experiments, when $k < .3$ all methods trigger higher standard and make short-term accuracy low).

The social impact on top-k policy. The Top-k strategy ultimately depends on a single score to determine classification results[1]. Consequently, users are driven to optimize a single "golden standard," pushing the model toward stricter decision boundaries and reducing overall diversity. In our experiments, this reliance on a single metric not only causes model shifts towards a higher standard but can also generate social challenges such as user stress and anxiety (Halko and Sääksvuori 2017; Feri, Innocenti, and Pin 2013). Recent research from Purdue University indicates that an increasing number of content creators experience burnout or stop creating content on platforms like YouTube due to the competitive and stressful environment (Thorne 2023).

On the other hand, the cost of features is crucial in algorithmic recourse, yet it is rarely considered in model fitting and evolution. Although strategic learning (Hardt et al. 2015; Levanon and Rosenfeld 2021) addresses this issue, its primary focus is often single-step accuracy rather than long-term dynamics. In reality, the cost function determines the direction of the data distribution's shift, so inferring feature costs can help predict the trajectory of model evolution and even steer it intentionally. Additionally, the semantic meaning of features may play a significant role. For instance, should a video's predicted quality rely more on its content-related characteristics or on socially driven metrics such as the number of likes? After all, careful system design and monitoring are probably essential to mitigate unintended consequences and ensure long-term stability and fairness (Bell et al. 2024). Finally, we note that even if the cost function is unmeasurable or acts as a black box, one can still design strategies—such as the Fair-Top-k approach—that accept diverse points and mitigate the pitfalls of focusing on a single score.

Acknowledgments

Jie Gao would like to acknowledge funding support from NSF through CCF-2118953, DMS-2311064, DMS-2220271, IIS-2229876, and CNS-2515159. Hao-Tsung Yang would like to acknowledge funding support from NSTC through NSTC-114-2221-E-008-049 and NSTC-113-2221-E-008-086.

References

Altmeyer, P.; Angela, G.; Buszydlik, A.; Dobiczek, K.; van Deursen, A.; and Liem, C. C. 2023. Endogenous macrodynamics in algorithmic recourse. In *2023 IEEE Conference*

[1] In practice, even with complex deep learning models that consider multiple objectives, these objectives are often combined into one mixed objective function (e.g., via Lagrangian relaxation) to ensure differentiability.

on Secure and Trustworthy Machine Learning (SaTML), 418–431. IEEE.

Arrieta, A. B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion*, 58: 82–115.

Barocas, S.; Hardt, M.; and Narayanan, A. 2017. Fairness in machine learning. *Nips tutorial*, 1: 2017.

Bastani, O.; Kim, C.; and Bastani, H. 2018. Interpretability via Model Extraction. arXiv:1706.09773.

Bell, A.; Fonseca, J.; Abrate, C.; Bonchi, F.; and Stoyanovich, J. 2024. Fairness in Algorithmic Recourse Through the Lens of Substantive Equality of Opportunity. arXiv:2401.16088.

Bell, A.; Fonseca, J.; and Stoyanovich, J. 2024. The Game Of Recourse: Simulating Algorithmic Recourse over Time to Improve Its Reliability and Fairness. In *SIGMOD Conference Companion*, 464–467.

Bertossi, L. 2020. An ASP-based approach to counterfactual explanations for classification. In *Rules and Reasoning: 4th International Joint Conference, RuleML+ RR 2020, Oslo, Norway, June 29–July 1, 2020, Proceedings 4*, 70–81. Springer.

Bikker, J. A.; and Haaf, K. 2002. Competition, concentration and their relationship: An empirical analysis of the banking industry. *Journal of banking & finance*, 26(11): 2191–2214.

Brodersen, K. H.; Ong, C. S.; Stephan, K. E.; and Buhmann, J. M. 2010. The Balanced Accuracy and Its Posterior Distribution. In *2010 20th International Conference on Pattern Recognition*, 3121–3124. Istanbul, Turkey: IEEE.

Bui, N.; Nguyen, D.; Yue, M.-C.; and Nguyen, V. A. 2025. Coverage-Validity-Aware Algorithmic Recourse. arXiv:2311.11349.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr., P. H. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. *Proceedings of the European Conference on Computer Vision*, 532–547.

Chen, Y.; Wang, J.; and Liu, Y. 2023. Learning to Incentivize Improvements from Strategic Agents. *Transactions on Machine Learning Research*.

Chitra, U.; and Musco, C. 2020. Analyzing the impact of filter bubbles on social network polarization. In *Proceedings of the 13th international conference on web search and data mining*, 115–123. New York, NY, USA: Association for Computing Machinery.

De Toni, G.; Teso, S.; Lepri, B.; and Passerini, A. 2025. Time Can Invalidate Algorithmic Recourse. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, 89–107. ACM.

Estornell, A.; Chen, Y.; Das, S.; Liu, Y.; and Vorobeychik, Y. 2023. Incentivizing Recourse through Auditing in Strategic Classification. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 400–408. International Joint Conferences on Artificial Intelligence Organization.

Ezrachi, A.; and Stucke, M. E. 2015. The curious case of competition and quality. *Journal of Antitrust Enforcement*, 3(2): 227–257.

Feri, F.; Innocenti, A.; and Pin, P. 2013. Is there psychological pressure in competitive environments? *Journal of Economic Psychology*, 39: 249–256.

Fiok, K.; Farahani, F. V.; Karwowski, W.; and Ahram, T. 2022. Explainable artificial intelligence for education and training. *The Journal of Defense Modeling and Simulation*, 19(2): 133–144.

Fokkema, H.; Garreau, D.; and van Erven, T. 2024. The Risks of Recourse in Binary Classification. In *International Conference on Artificial Intelligence and Statistics*, 550–558. PMLR.

Fonseca, J.; Bell, A.; Abrate, C.; Bonchi, F.; and Stoyanovich, J. 2023. Setting the right expectations: Algorithmic recourse over time. In *Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*. New York, NY, USA: Association for Computing Machinery.

Gao, R.; and Lakkaraju, H. 2023. On the impact of algorithmic recourse on social segregation. In *International Conference on Machine Learning*, 10727–10743. PMLR.

Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; and Pedreschi, D. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5): 1–42.

Gunning, D.; Stefik, M.; Choi, J.; Miller, T.; Stumpf, S.; and Yang, G.-Z. 2019. XAI—Explainable artificial intelligence. *Science robotics*, 4(37): eaay7120.

Guo, H.; Jia, F.; Chen, J.; Squicciarini, A.; and Yadav, A. 2023. RoCourseNet: Robust Training of a Prediction Aware Recourse Model. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*, 619 – 628. Birmingham, United Kingdom: ACM.

Gupta, V.; Nokhiz, P.; Roy, C. D.; and Venkatasubramanian, S. 2019. Equalizing Recourse across Groups. arXiv:1909.03166.

Halko, M.-L.; and Sääksvuori, L. 2017. Competitive behavior, stress, and gender. *Journal of Economic Behavior & Organization*, 141: 96–109.

Harari, Y. N. 2024. *Nexus: A brief history of information networks from the Stone Age to AI*. Signal.

Hardt, M.; Megiddo, N.; Papadimitriou, C.; and Wootters, M. 2015. Strategic classification. *CoRR*, abs/1506.06980.

Hardt, M.; and Mendler-Dünner, C. 2025. Performative Prediction: Past and Future. arXiv:2310.16608.

Hennig-Thurau, T.; Marchand, A.; and Marx, P. 2012. Can automated group recommender systems help consumers make better choices? *Journal of Marketing*, 76(5): 89–109.

Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; and Riedl, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1): 5–53.

Karimi, A.-H.; Barthe, G.; Schölkopf, B.; and Valera, I. 2022. A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. *ACM Computing Surveys*, 55(5): 1–29.

Khosravi, H.; Shum, S. B.; Chen, G.; Conati, C.; Tsai, Y.-S.; Kay, J.; Knight, S.; Martinez-Maldonado, R.; Sadiq, S.; and Gašević, D. 2022. Explainable artificial intelligence in education. *Computers and Education: Artificial Intelligence*, 3: 100074.

Kingma, D. P. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirdemir, B.; Kready, J.; Mead, E.; Hussain, M. N.; and Agarwal, N. 2021. Examining video recommendation bias on YouTube. In *International Workshop on Algorithmic Bias in Search and Recommendation*, 106–116. Springer.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; Clopath, C.; Kumaran, D.; and Hadsell, R. 2017. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526.

Klug, D.; Qin, Y.; Evans, M.; and Kaufman, G. 2021. Trick and please. A mixed-method study on user assumptions about the TikTok algorithm. In *Proceedings of the 13th ACM Web Science Conference 2021*, 84–92. New York, NY, USA: Association for Computing Machinery.

Levanon, S.; and Rosenfeld, N. 2021. Strategic classification made practical. In *International Conference on Machine Learning*, 6243–6253. PMLR.

Liang, Z.; Zhang, G.; Huang, J. X.; and Hu, Q. V. 2014. Deep learning for healthcare decision making with EMRs. In *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 556–559. IEEE.

Lin, J. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1): 145–151.

Lopez-Paz, D.; and Ranzato, M. 2017a. Gradient Episodic Memory for Continual Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Lopez-Paz, D.; and Ranzato, M. 2017b. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6470–6479. Red Hook, NY, USA: Curran Associates Inc.

MacDonald, T. W. 2023. "How it actually works": Algorithmic lore videos as market devices. *New Media & Society*, 25(6): 1412–1431.

McAfee, R. P.; Mialon, H. M.; and Williams, M. A. 2004. What is a Barrier to Entry? *American Economic Review*, 94(2): 461–465.

Nguyen, D.; Bui, N.; and Nguyen, V. A. 2023. Feasible Recourse Plan via Diverse Interpolation. In *International Conference on Artificial Intelligence and Statistics*, 4679–4698. PMLR.

O'Brien, A.; and Kim, E. 2021. Multi-Agent Algorithmic Recourse. arXiv:2110.00673.

Pehrsson, A. 2009. Barriers to entry and market strategy: a literature review and a proposed model. *European Business Review*, 21(1): 64–77.

Poyiadzi, R.; Sokol, K.; Santos-Rodriguez, R.; De Bie, T.; and Flach, P. 2020. FACE: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 344–350.

Qin, H.; Gong, R.; Liu, X.; Bai, X.; Song, J.; and Sebe, N. 2020. Binary neural networks: A survey. *Pattern Recognition*, 105: 107281.

Rawal, K.; Kamar, E.; and Lakkaraju, H. 2021. Algorithmic Recourse in the Wild: Understanding the Impact of Data and Model Shifts. arXiv:2012.11788.

Rawal, R.; Cesa-Bianchi, N.; Saligrama, V.; and Goyal, P. 2023. Causal Algorithmic Recourse under Recourse-induced Distributional Shift. *arXiv preprint arXiv:2306.08528*.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5533–5542.

Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T. P.; and Wayne, G. 2019. Experience Replay for Continual Learning. In *Advances in Neural Information Processing Systems*, volume 32.

Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; and Hadsell, R. 2016. Progressive Neural Networks. *arXiv preprint arXiv:1606.04671*.

Shankar, D.; Narumanchi, S.; Ananya, H. A.; Kompalli, P.; and Chaudhury, K. 2017. Deep Learning based Large Scale Visual Recommendation and Search for E-Commerce. *CoRR*, abs/1703.02344.

Shao, X.; and Kersting, K. 2022. Gradient-based Counterfactual Explanations using Tractable Probabilistic Models. arXiv:2205.07774.

Shi, H.; Xu, Z.; Wang, H.; Qin, W.; Wang, W.; Wang, Y.; Wang, Z.; Ebrahimi, S.; and Wang, H. 2024. Continual Learning of Large Language Models: A Comprehensive Survey. arXiv:2404.16789.

Thorne, S. 2023. # Emotional: Exploitation & Burnout in Creator Culture. *CLCWeb: Comparative Literature and Culture*, 24(4): 7.

Upadhyay, S.; Joshi, S.; and Lakkaraju, H. 2021a. On Minimizing the Impact of Dataset Shifts on Actionable Explanations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 478–488. ACM.

Upadhyay, S.; Joshi, S.; and Lakkaraju, H. 2021b. Towards robust and reliable algorithmic recourse. *Advances in Neural Information Processing Systems*, 34: 16926–16937.

Ustun, B.; Spangher, A.; and Liu, Y. 2019. Actionable recourse in linear classification. In *Proceedings of the conference on fairness, accountability, and transparency*, 10–19.

Venkatasubramanian, S.; and Alfano, M. 2020. The philosophical basis of algorithmic recourse. In *Proceedings of*

the 2020 conference on fairness, accountability, and transparency, 284–293. New York, NY, USA: Association for Computing Machinery.

Verma, S.; Boonsanong, V.; Hoang, M.; Hines, K. E.; Dickerson, J. P.; and Shah, C. 2020. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Comput. Surv.*, 56(12).

Wachter, S.; Mittelstadt, B.; and Russell, C. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.*, 31: 841.

Yadav, P.; Hase, P.; and Bansal, M. 2022. Low-Cost Algorithmic Recourse for Users With Uncertain Cost Functions. arXiv:2111.01235.

Yeh, I.-C.; and Lien, C.-h. 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications*, 36(2): 2473–2480.

Yesilada, M.; and Lewandowsky, S. 2022. Systematic review: YouTube recommendations and problematic content. *Internet policy review*, 11(1).

Yoon, J.; Yang, E.; Lee, J.; and Hwang, S. J. 2018. Lifelong Learning with Dynamically Expandable Networks. In *International Conference on Learning Representations (ICLR)*.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual Learning Through Synaptic Intelligence. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Zhang, S.; Yao, L.; Sun, A.; and Tay, Y. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1): 1–38.

Zhou, L. 2020. Product advertising recommendation in e-commerce based on deep learning and distributed expression. *Electronic Commerce Research*, 20(2): 321–342.