A BETTER FIT NEED NOT BE A BETTER POLICY: RETRAINING IN PERSISTENT GRAPH ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph-based predictions can alter the features and outcomes used for later training. We formulate this interaction as a stateful graph-feedback process and compare repeated refitting with continued deployment of the initial classifier. A common-state comparison isolates the contribution of refitting, while a policy comparison also captures the contribution of the graph states generated by deployment.

We instantiate the framework in two settings. *(i)* For synthetic node classification, we establish well-posedness and existence of an invariant distribution, and derive a finite-horizon condition under which lower loss on common states transfers to lower policy loss. Experiments across independently generated synthetic datasets show that fitting and induced-state contributions can reinforce or offset one another. An end-to-end study further shows that the update policy can reverse the performance ordering of GNN architectures. *(ii)* In a user–product graph simulation, explanation recourse changes product descriptions while historical rating labels remain fixed. Most implemented description edits continue to place the corresponding products above the eligibility threshold after refitting, while aggregate loss changes remain small. These results show that a better fit on an updated graph state may yield a worse deployment policy.

1 INTRODUCTION

Graph-based classifiers combine an entity's features with information from connected entities. Graph neural networks (GNNs), for example, aggregate neighboring features to predict node labels (Kipf & Welling, 2017). When these predictions guide decisions, deployment can change the features used for prediction or the outcomes being predicted. Later training and evaluation then depend on earlier deployments.

Consider a user–product graph used to predict favorable ratings. A product with a low score may receive recourse in the form of a fact-preserving description edit intended to raise its predicted favorability for a target audience. Implementing the edit changes the product representation and, through graph aggregation, predictions for connected user–product pairs. Refitting on the updated graph can then change subsequent predictions and recourse recommendations. This sequence of prediction, graph response, and refitting creates a feedback loop on a persistent graph (Eilat & Rosenfeld, 2023; König et al., 2025).

Performative prediction on a persistent graph. This example exposes two defining features of graph feedback. First, a local response can alter predictions for other units through graph aggregation. Second, when the same units persist, each deployment acts on a graph state that retains the effects of earlier deployments. Refitting adapts the classifier to the updated state, and its predictions can induce the next state transition. This raises our central question: *When does repeated refitting yield lower predictive loss than continued deployment of the initial classifier?*

We answer this question through two complementary comparisons. First, we evaluate the refitted and initial classifiers on the same graph state, which isolates the contribution of refitting. Second, we evaluate each policy along the state trajectory it induces, which also captures the consequences of its earlier deployments. Together, the comparisons distinguish changes in predictions from changes in the states on which those predictions are evaluated.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Our contributions. (I) A framework for refitting under graph feedback. We formulate deployment as a stateful graph process and define common-state and policy-induced-trajectory comparisons for the refitting and fixed-classifier policies. **(II) Guarantees for the synthetic process.** We establish well-posed updates and existence of an invariant distribution, then derive a finite-horizon condition under which a common-state advantage implies lower policy loss. **(III) Evidence from two graph-feedback settings.** Synthetic experiments show that fitting and induced-state contributions can reinforce or offset one another, and an end-to-end study shows that the preferred GNN architecture can depend on the update policy. A user–product recourse study separately evaluates predictive loss and whether products with implemented description edits remain above the threshold after refitting.

2 Graph Feedback: Framework and Instances

We first define the state, transition, and update policies shared by both settings. We then instantiate these objects for synthetic node classification and explanation recourse on a user–product graph.

Let $s_t = (\mathcal{G}_t, \boldsymbol{y}_t) \in \mathcal{S}$ denote the environment state at round t, where $\mathcal{G}_t$ records the graph and its response-relevant attributes and $\boldsymbol{y}_t$ records current outcomes. The prediction units, data split, and other fixed design choices remain unchanged across rounds. A deployed classifier produces a signal or action $\boldsymbol{u}_t$—the node probability vector in the synthetic instance and a recourse recommendation in the application—after which

$$s_{t+1} \mid (s_t, \boldsymbol{u}_t) \sim P(\cdot \mid s_t, \boldsymbol{u}_t). \tag{1}$$

The current state carries the effects of earlier deployments. Thus, deploying the same classifier from different states can produce different next-state distributions. Unlike population-distribution formulations of stateful performative prediction (Brown et al., 2022), the synthetic state here is one joint configuration of a persistent finite graph whose units are coupled in prediction and outcome evolution.

Let $O(s_t)$ contain the graph and labels available for fitting, and let $\mathcal{T}$ be the fitting rule. Starting from the same state and initial fit $\boldsymbol{\theta}_0 = \mathcal{T}(O(s_0))$, we compare

$$\boldsymbol{\theta}^{\pi}_{t+1} = \begin{cases} \mathcal{T}(O(s^{\pi}_{t+1})), & \pi = \text{refit}, \\ \boldsymbol{\theta}_0, & \pi = \text{fixed}. \end{cases} \tag{2}$$

For mean held-out loss $\mathcal{R}(f, s)$ and $0 \le t_0 < H$, define

$$J_{t_0,H}(\pi) \coloneqq \frac{1}{H - t_0} \sum_{t=t_0+1}^{H} \mathbb{E}_{\pi}\left[\mathcal{R}(f_{\boldsymbol{\theta}^{\pi}_t}, s^{\pi}_t)\right]. \tag{3}$$

Each policy is therefore evaluated on the states generated by its own deployments.

Why the graph state matters. Equation (1) describes a stateful performative transition in which the state contains the attributed graph and its outcome vector. Graph structure couples the prediction units: a node's prediction can depend on neighboring features, its response can alter features used in later predictions for other nodes, and its outcome can depend on neighboring responses. The fitting rule observes current outcomes only on the training subset, whereas Equation (3) evaluates held-out units after the graph responds. A distribution over s_t therefore represents uncertainty over joint graph configurations rather than over independent examples. Consequently, a better fit on one common state need not order policies that generate different future graph states.

Table 1: Instances of the graph-feedback framework.

	Synthetic node classification	Explanation recourse
Units	Graph nodes	User–product pairs
Action	Node probability vector	Recourse recommendation
Changing state	Mutable features and current outcomes	Descriptions and product features
Fixed components	Nodes, edges, immutable features	Graph and historical rating labels
Evaluation	Current test loss	Test loss and post-refit threshold retention

2.1 SYNTHETIC NODE CLASSIFICATION

We use a fixed graph with a transductive train/validation/test split. Each node has an immutable feature, a mutable feature, and a current binary outcome. Raw features (R), graph-propagated features (G), and their concatenation (C) feed a ridge-regularized logistic readout. Deployment moves each mutable feature toward a prediction-dependent target. To make this response explicit, write $\boldsymbol{X}_t = [\,\mathbf{r} \mid \boldsymbol{x}_t\,]$, let $p_{i,t}$ be node i's predicted probability, and set

$$x_{i,t+1} = (1-a)x_{i,t} + a[b_i + \chi c(2p_{i,t} - 1)]. \tag{4}$$

Here $b_i = x_{i,0}$, $a \in [0,1]$ sets the adjustment rate, $c > 0$ bounds the displacement, and $\chi \in \{-1,+1\}$ determines its direction. With $\Delta x_{j,t} = x_{j,t+1} - x_{j,t}$, the probability used when node i's outcome is resampled is

$$q_{i,t} = \sigma\big(x_{i,t+1} + \mathrm{mean}_{j\in\mathcal{N}(i)}\, x_{j,t+1} + \mathrm{mean}_{j\in\mathcal{N}(i)}\, \Delta x_{j,t}\big)\,. \tag{5}$$

The next outcome follows

$$y_{i,t+1} \sim \mathrm{Bernoulli}\,\big((1-\varepsilon)y_{i,t} + \varepsilon q_{i,t}\big). \tag{6}$$

Thus, the current outcome is retained with probability $1-\varepsilon$ and otherwise resampled from conditions at the node and its neighbors. The refitting policy uses the resulting training labels; the fixed-classifier policy retains the initial coefficients. Appendix B.1 specifies the representation, fitting, and evaluation equations.

This construction introduces cross-unit dependence at two stages. Feature propagation makes each prediction depend on neighboring nodes, while the neighbor-dependent outcome probabilities connect their subsequent outcomes. Refitting observes current outcomes only on the fixed training subset. The evolving object is therefore one partially observed, dependent graph state rather than a new collection of independent examples at each round.

2.2 EXPLANATION RECOURSE ON A USER–PRODUCT GRAPH

The second instance predicts whether specified user–product pairs have favorable historical ratings. Training interactions define a fixed bipartite graph. For each product, the platform averages predictions over a fixed audience and recommends the least-cost permitted description whose score crosses a fixed eligibility threshold. Implemented description edits change product features, graph-propagated pair features, and later recourse recommendations. The historical graph edges and rating labels remain fixed. The simulated response is limited to description changes; future purchases and ratings remain unmodeled. Appendix B.2 specifies the predictor, recourse-recommendation rule, and state transition.

More precisely, if A_i is product i's audience and $\boldsymbol{\psi}_{ui}(\mathbf{X}^{i,k})$ is the pair representation under candidate description k, its score under fitted coefficients $\mathbf{v}_j$ is

$$g_i^j(k) \coloneqq \frac{1}{|A_i|}\sum_{u\in A_i} \sigma\big(\boldsymbol{\psi}_{ui}(\mathbf{X}^{i,k})^\top \mathbf{v}_j\big)\,. \tag{7}$$

A product below threshold τ selects the least-cost permitted k with $g_i^j(k) \geq \tau$, using fixed tie-breaking. The selected description is the recourse recommendation. The corresponding description edits are implemented simultaneously before features are recomputed and the next fit is obtained. Post-refit threshold retention is the proportion of products whose implemented description edits remain at or above τ after refitting.

The two instances expose complementary forms of feedback. The synthetic process closes the loop through mutable features and current outcomes, allowing us to study a persistent stochastic graph process. The recourse instance holds historical edges and labels fixed and isolates feedback through implemented description edits and subsequent refitting. The same common-state and policy comparisons apply in both settings, while each setting retains its own response assumptions.

3 ANALYSIS

The analysis proceeds in four stages. For the synthetic process, we first establish well-posedness and existence of an invariant distribution. We then separate fitting from induced-state contributions and

derive a condition under which a common-state advantage implies lower policy loss. Finally, for the recourse instance, we give a sufficient condition for refitting to leave the recommended description unchanged.

3.1 PROCESS PROPERTIES

Let n be the number of nodes, let b_i be node i's initial mutable feature, and let c bound its prediction-dependent displacement. Define

$$\mathcal{C} \coloneqq \prod_{i=1}^{n} [b_i - c, b_i + c], \qquad B_w \coloneqq \sqrt{2 \log 2/\lambda}. \tag{8}$$

Proposition 1 (Well-posed updates)**.** *For finite initial features, a nonempty training set, $a \in [0, 1]$, $c > 0$, and ridge penalty $\lambda > 0$: (i) the feature update preserves $\boldsymbol{x}_t \in \mathcal{C}$; (ii) for every admissible current data snapshot, the regularized training objective has a unique exact minimizer; and (iii) every fitted coefficient vector has norm at most B_w.*

Appendix C.1 proves the three claims. Regularizing every coefficient, including the intercept, also covers rounds whose observed training labels contain only one class.

The exact fit is a continuous function of the mutable features for every fixed training-label vector. Hence the reduced state $s_t = (\boldsymbol{x}_t, \boldsymbol{y}_t)$ evolves on the compact space $\mathcal{H} \coloneqq \mathcal{C} \times \{0, 1\}^n$ through a Feller transition kernel P.

Proposition 2 (Existence of an invariant distribution)**.** *Suppose the representation map $\boldsymbol{X} \mapsto \mathbf{Z}(\boldsymbol{X})$ is continuous. For every fixed dataset and parameters satisfying Proposition 1, and every $\varepsilon \in [0, 1]$, the exact-refitting process on $\mathcal{H}$ has an invariant probability distribution.*

Appendix C.2 contains the proof. This result shows that the persistent stochastic process admits at least one stationary regime. It makes no contraction assumption; uniqueness and convergence from every initial state require additional conditions. When $\varepsilon > 0$, every node changes its outcome infinitely often almost surely (Proposition 4 in the appendix).

3.2 SEPARATING FITTING AND INDUCED-STATE CONTRIBUTIONS

At every round, inserting the initial classifier on the refitting state yields the exact decomposition

$$\begin{aligned} &\mathcal{R}(f_t^{\mathrm{refit}}, s_t^{\mathrm{refit}}) - \mathcal{R}(f_0, s_t^{\mathrm{fixed}}) \\ &= \underbrace{\mathcal{R}(f_t^{\mathrm{refit}}, s_t^{\mathrm{refit}}) - \mathcal{R}(f_0, s_t^{\mathrm{refit}})}_{\text{fitting contribution on a common state}} + \underbrace{\mathcal{R}(f_0, s_t^{\mathrm{refit}}) - \mathcal{R}(f_0, s_t^{\mathrm{fixed}})}_{\text{induced-state contribution}}. \end{aligned} \tag{9}$$

The allocation depends on the initial classifier used as the reference, while the identity itself holds for every graph operator and response mechanism. As a baseline, consider the matched setting with no feature response ($a = 0$). When the initial outcomes follow the stationary baseline mechanism, the classifier cannot alter the states encountered by the two policies, and their expected losses are equal at every round. Appendix D.1 states and proves this result.

3.3 WHEN A COMMON-STATE ADVANTAGE TRANSFERS TO POLICY PERFORMANCE

Consider two representations A and B fitted exactly at every round under the synthetic transition. They share the graph, training and test sets, ridge penalty, response parameters, initial state, and coupled outcome randomness. Assume A's representation vectors are uniformly bounded and Lipschitz in the mutable features. Let

$$J_T(A) \coloneqq T^{-1} \sum_{t=1}^{T} \mathbb{E}\mathcal{R}_A(s_t^A), \qquad \Gamma_T \coloneqq T^{-1} \sum_{t=1}^{T} \mathbb{E}[\mathcal{R}_B(s_t^B) - \mathcal{R}_A(s_t^B)].$$

Here Γ_T measures A's average advantage when both representations are refitted and evaluated on B's states.

This common-state advantage alone is insufficient to order the two deployment policies: A and B can change subsequent features and outcomes in different ways and therefore be evaluated on different future states. The purpose of the next result is to quantify how large the common-state advantage must be to dominate the loss change caused by this trajectory separation.

To bound the difference between the states induced by the two representations, let $\mathbf{h}_A(s)$ and $\mathbf{h}_B(s)$ denote their fitted logit vectors on state s, and define

$$\boldsymbol{d}_t \coloneqq \begin{bmatrix} \mathbb{E}\|\boldsymbol{x}_t^A - \boldsymbol{x}_t^B\|_\infty \\ \mathbb{E}\|\boldsymbol{y}_t^A - \boldsymbol{y}_t^B\|_1/n \end{bmatrix}, \qquad \nu_t \coloneqq \mathbb{E}\|\mathbf{h}_A(s_t^B) - \mathbf{h}_B(s_t^B)\|_\infty.$$

Fitting sensitivity and the coupled state transition yield nonnegative $\mathbf{B}$, $\mathbf{g}$, and $\mathbf{c}_*$ such that

$$\boldsymbol{d}_{t+1} \le \mathbf{B}\boldsymbol{d}_t + \mathbf{g}\nu_t, \qquad |\mathbb{E}[\mathcal{R}_A(s_t^A) - \mathcal{R}_A(s_t^B)]| \le \mathbf{c}_*^\top \boldsymbol{d}_t. \tag{10}$$

The explicit forms of $\mathbf{B}$, $\mathbf{g}$, and $\mathbf{c}_*$ in Equation (35) depend on the response parameters, ridge penalty, training-set size, and representation bounds. Since the policies start from the same state, $\boldsymbol{d}_0 = \mathbf{0}$. Iterating Equation (10) therefore defines

$$C_T \coloneqq \frac{1}{T}\sum_{t=1}^{T} \mathbf{c}_*^\top \sum_{k=0}^{t-1} \mathbf{B}^{t-1-k}\mathbf{g}\nu_k. \tag{11}$$

Theorem 1 (Preservation of a common-state advantage)**.** *Under the preceding assumptions, the coupling bound C_T in Equation* (11) *satisfies*

$$|J_T(A) - J_T(B) + \Gamma_T| \le C_T. \tag{12}$$

Consequently, $\Gamma_T > C_T$ implies $J_T(A) < J_T(B)$.

Implications. *(i)* The theorem yields a sufficient transfer condition: a common-state advantage becomes a policy-level advantage when it exceeds the trajectory term C_T. For example, if $\Gamma_T = 0.05$ and $C_T = 0.02$, then $J_T(A) - J_T(B) \le -0.03$, so A retains an advantage of at least 0.03. *(ii)* The result is finite-horizon and applies before convergence to an invariant distribution. It therefore applies while the policies are still changing the states they encounter. *(iii)* The condition is sufficient and may be conservative. Since C_T uses uniform sensitivity bounds, A may still have lower policy loss when $\Gamma_T \le C_T$. *(iv)* The no-feature-response case provides an informative limit. When $a = 0$, the feedback coefficient $\eta = ac/2$ and hence $\mathbf{g}$ are zero, so $C_T = 0$. The two representations then encounter the same states under the coupling, and their common-state comparison directly orders their policy losses. *(v)* With feedback, the bound identifies the quantities that can prevent this transfer. The sequence ν_k measures how differently A and B represent the same state; a, c, and ε enter through the state response; and the ridge penalty and training-set size enter through fitting sensitivity. For example, weaker regularization enlarges the displayed sensitivity constants in Appendix D. The relevant comparison therefore places the common-state margin Γ_T against the accumulated trajectory term C_T; one-step fitted losses alone are insufficient.

The theorem and the empirical decomposition serve complementary purposes. The theorem supplies a sufficient condition that can be evaluated from sensitivity bounds before comparing the realized policy trajectories. Equation (9) then computes the exact attribution after both trajectories have been generated.

3.4 Threshold robustness of recourse recommendations

For product i and candidate description k, define the coefficient-change bound

$$b_i(k) \coloneqq \frac{1}{4|A_i|}\sum_{u\in A_i} \left|\boldsymbol{\psi}_{ui}(\mathbf{X}^{i,k})^\top(\mathbf{v}_{\text{refit}} - \mathbf{v}_{\text{fixed}})\right|.$$

Proposition 3 (A sufficient condition for an unchanged recourse recommendation)**.** *Fix the representation, its scaling, and the current state. If $|g_i^{\text{fixed}}(k) - \tau| > b_i(k)$ for the current description and every permitted candidate, then the fixed and refitted classifiers recommend the same description, provided costs, permissions, and tie-breaking are unchanged.*

The condition follows from the sigmoid's $1/4$-Lipschitz constant and accommodates the nonsmooth discrete recourse-recommendation map. Appendix H.1 contains the proof. Appendix H also gives signed loss bounds and records why deterministic finite-candidate trajectories eventually cycle.

4 EXPERIMENTAL EVALUATION

Our evaluation asks three questions. *(i)* Does repeated refitting reduce loss along the trajectory it induces? *(ii)* How much of the policy gap comes from fitting on a common state and how much comes from induced states? *(iii)* In the recourse setting, do products with implemented description edits remain above the eligibility threshold after refitting? We examine these questions in two complementary studies. The synthetic study evaluates repeated refitting across independently generated graphs and repeated outcome realizations. The application combines real user–product interactions with simulated adoption of reviewed description edits.

4.1 SYNTHETIC NODE CLASSIFICATION

We generate 400-node graphs with binary latent groups, noisy immutable and mutable features, and fixed 60/20/20 train/validation/test memberships. The R/G/C representations use raw, two-hop propagated, or combined features. The primary setting uses ridge penalty $\lambda = 0.1$, feature adjustment $a = 0.1$, displacement $c = 1$, and outcome-resampling probability $\varepsilon = 0.1$. A matched $a = 0$ setting removes feature response while retaining outcome randomness.

We generate twenty independent datasets and five coupled outcome realizations per dataset. Three representations, two policies, and two response conditions produce 1,200 trajectories of 1,000 rounds. The primary summary averages rounds 601–1000. We first average the five realizations within each dataset and then form paired dataset-bootstrap intervals; rounds and realizations on one graph remain within the same dependent group. Appendix E specifies the generator, baseline comparisons, and complete settings.

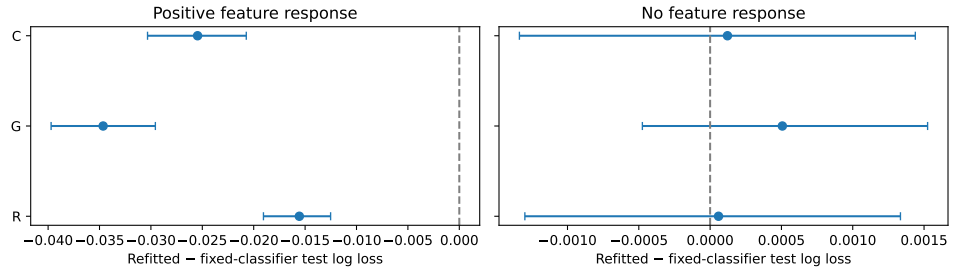

Figure 1: Refitted-minus-fixed synthetic test loss with pointwise 95% paired dataset-bootstrap intervals. Negative values favor refitting. Under the $a = 0$ baseline, the expected policy gap is zero, while a finite collection of generated datasets may have a nonzero empirical mean.

Refitting improves loss under positive feature response. Under positive feature feedback, the mean refitted-minus-fixed log-loss gaps are -0.015560 for R, -0.034648 for G, and -0.025459 for C, with 95% intervals $[-0.019052, -0.012519]$, $[-0.039708, -0.029566]$, and $[-0.030328, -0.020722]$, respectively. Refitted C also improves on G and R on every dataset average. The no-feature-response gaps are small and their intervals include zero, as predicted by Proposition 5.

Induced states can offset or reinforce a fitting advantage. The decomposition in Equation (9) shows why the policy comparison adds information. Refitted C minus G has fitting and induced-state contributions -0.043501 and $+0.007318$, respectively, so the induced states offset part of the fitting advantage. For C minus R the corresponding values are -0.038554 and -0.039518, so the contributions reinforce one another. Appendix E.5 shows that the paired policies continue to differ in mutable features and outcomes throughout the evaluation window.

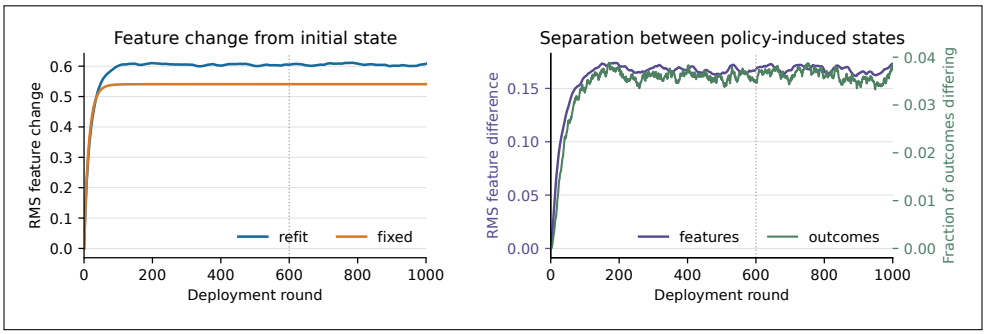

Figure 2: State movement for C under positive feature response. The left panel shows displacement from the common initial state. The right panel shows separation between the states induced by refitting and keeping the initial classifier fixed. Curves average five coupled outcome realizations within each of twenty independently generated datasets and then average equally across datasets. The dotted line marks the start of the evaluation window.

The trajectory separation in Figure 2 arises under the mixed outcome mechanism. To determine whether C's advantage depends on combining own-node and neighbor information, we repeat the comparison under matched own-only and neighbor-only mechanisms. In each endpoint mechanism, C and the corresponding single-channel representation are statistically indistinguishable. The primary advantage therefore occurs when both channels carry complementary information for the outcome update.

4.2 End-to-end GNN robustness

The primary study isolates feedback using fixed graph representations and a convex logistic readout. We next test whether its empirical conclusions extend to end-to-end MLP, GCN, and skip-GCN predictors. These experiments use the same twenty datasets and paired policy comparisons, while optimizing nonconvex neural objectives. The guarantees in Section 3 continue to apply only to the convex readouts.

Table 2: Late-window log loss under the mixed outcome mechanism and positive feature response. The final two columns decompose the refitted-minus-fixed gap on a common state.

Predictor	Fixed	Refitted	Gap	Fitting	Induced state
MLP	0.5045	0.5011	−0.0034	−0.0043	+0.0010
GCN	0.5276	0.4737	−0.0539	−0.0159	−0.0380
Skip-GCN	0.4931	0.4973	+0.0043	+0.0021	+0.0022

Refitting reduces GCN loss by 0.05390, with paired 95% interval $[-0.07200, -0.03832]$; both the fitting and induced-state contributions are favorable. The MLP and skip-GCN policy gaps are smaller and their intervals contain zero. Removing feature response also makes all three gaps inconclusive, locating the GCN improvement in the prediction-to-feature feedback pathway.

The update policy reverses the graph-model ordering. Under the fixed policy, skip-GCN minus GCN loss is -0.03453 with interval $[-0.05285, -0.01579]$; under refitting it is $+0.02364$ with interval $[0.01194, 0.03503]$. The paired interaction is $+0.05817$ with interval $[0.04131, 0.07664]$. Thus, the architecture with lower loss under the fixed-classifier policy may have higher loss under repeated refitting. Appendix F reports training details, sensitivity analyses, and secondary metrics. These results are empirical. The convex guarantees remain specific to the analyzed readouts.

What the synthetic evidence establishes. Three comparisons support distinct conclusions. *(i)* Under the no-feature-response setting, the expected policy gap is zero by Proposition 5, and every empirical interval includes zero. The systematic gaps under positive response therefore arise through the prediction-to-feature pathway. The exact zero-gap result excludes outcome innovations alone

as their source. *(ii)* Under feedback, the decomposition distinguishes a fitting advantage from its downstream consequences. The induced-state contribution offsets part of C's advantage over G and reinforces its advantage over R, showing that this contribution has either sign. *(iii)* The end-to-end study extends the empirical pattern beyond convex readouts: skip-GCN has lower loss than GCN when the initial classifier remains fixed; GCN has lower loss under repeated refitting. Architecture and update policy therefore require joint selection in this environment.

The synthetic studies evaluate how predictive models shape evolving features and outcomes. We next examine a different feedback pathway in which model predictions trigger description edits while historical rating labels remain fixed.

4.3 PRODUCT-DESCRIPTION RECOURSE

The application uses the Musical Instruments category of Amazon Reviews 2023 (Hou et al., 2026). Training interactions define the user–product graph, and a favorable label denotes a recorded rating of at least four stars. A frozen text encoder represents product descriptions, while user features summarize positive and negative training reviews and current neighboring-product features. R/G/C again denote raw, propagated, and combined product channels; each uses a ridge-regularized logistic readout.

Eligibility is the mean predicted favorable-rating probability over a fixed audience of 100 customers. For each product below the threshold, the least-cost reviewed description that crosses the threshold becomes the recourse recommendation. All recommendations in a round are computed from the same state, and the corresponding description edits are then implemented simultaneously. Of 243 preselected products, 124 pass source review. The cleared warm cohort contains 1,820 test pairs; the full warm set contains 144,991. This is a specified simulation of full adoption. Actual Amazon deployment and seller adoption require separate evidence.

Table 3: Twenty-round recourse results. The post-refit threshold-retention column counts products whose first-round implemented description edit keeps their score above the threshold after the first refit. Loss gaps are refitted minus fixed in units of 10^{-6}; negative values favor refitting.

Model	Initial edits	Above τ after refit	Cleared cohort	Full warm
R	34	30	-37.797	5.204
G	32	32	1.105	-2.051
C	32	30	-30.692	0.815

Most edited products remain eligible after refitting. We run twenty deterministic rounds and permit a product to return to its original description if that description again crosses the threshold. Refitting produces three additional implemented description edits for R, none for G, and one for C. Subsequent changes cease quickly: every trajectory reaches an absorbing no-edit state by round three, and most first-round implemented description edits remain above the eligibility threshold after refitting. The loss gaps in Table 3 are several orders of magnitude smaller than the losses themselves: final full-warm losses are approximately 0.4182 for R, 0.4174 for G, and 0.4164 for C under both policies. On the full warm set, fitting slightly worsens R and C, while additional implemented description edits partly offset the change. For G, the state trajectories coincide, so fitting accounts for the complete policy gap. Since the loss gaps are small, we use signed empirical bounds to verify their reported directions for all final and average comparisons, including cold-inclusive evaluations. We use these bounds to characterize the realized evaluation samples; assessing sampling uncertainty requires an additional sampling model. We provide complete attributions and evaluation scopes in Appendix G.

Interpreting post-refit threshold retention and predictive loss. The post-refit threshold-retention rates are $30/34$ for R, $32/32$ for G, and $30/32$ for C, or approximately 88%, 100%, and 94%. These rates measure the robustness of discrete product-level decisions to the first refit. The loss gaps answer a different, population-level question. A localized set of implemented description edits can have little effect on an average over as many as 144,991 warm user–product pairs. Refitting can also change predictions for pairs involving products that receive no recourse recommendation.

High post-refit threshold retention and a negligible aggregate loss change are therefore compatible. Finally, the absorbing no-edit states reached by round three describe only this deterministic, finite-candidate simulation. Convergence of the stochastic outcome process studied in Section 3 remains unexamined.

Together, the two studies separate three questions: whether refitting improves predictions on a common state, whether it improves loss along the states it induces, and whether products with implemented description edits remain eligible after refitting. The answers need not agree.

5 Related Work

Performative prediction and graph feedback. Performative prediction distinguishes empirical fit from performance under model-induced distributions (Perdomo et al., 2020; Miller et al., 2021). Brown et al. (2022) study stateful transitions between population-level data distributions. Our setting retains one realized finite graph: fixed units are coupled through prediction and outcome evolution, and current labels are observed only on a training subset. We analyze finite-horizon loss without a contraction assumption on the joint graph-state transition.

Strategic learning and recourse. Graph-coupled strategic learning anticipates responses during classifier fitting (Eilat et al., 2023). Our study examines ordinary finite-sample refitting against continued deployment of the initial classifier. Performative recourse asks when recourse recommendations remain valid after deployment-induced distribution shift and model updates (König et al., 2025). Our application instead measures post-refit threshold retention and predictive loss separately while holding historical ratings fixed. Related work also studies strategic content creation in recommender systems (Eilat & Rosenfeld, 2023), fixed graph propagation (Wu et al., 2019), and decompositions of distribution-shift effects (Cai et al., 2026). Our decomposition attributes a policy gap to fitting and to the graph states generated by deployment.

6 Conclusion

We introduced common-state and policy-induced-trajectory comparisons for repeated refitting under graph feedback. In the synthetic study, induced-state changes can reinforce or offset a fitting advantage, and the preferred neural architecture changes with the update policy. In the recourse simulation, most products with implemented description edits remain eligible after refitting, while aggregate predictive differences are small. Threshold crossing, improved fit, and improved policy performance are therefore distinct outcomes.

Recommended evaluations. The analysis suggests three measurements for graph predictors in responsive environments: compare the updated and initial classifiers on the same post-deployment state; compare complete policies along their induced trajectories and report the fitting and induced-state contributions; and, when predictions trigger discrete interventions, report post-refit threshold retention separately from aggregate predictive loss. These measurements show whether an improvement arises from adaptation to the current graph, from the states produced by earlier deployments, or from both.

Limitations. The synthetic theory assumes exact convex refitting and a specified graph response. The application combines real interactions and text with simulated adoption, fixed historical labels, and fixed candidate sets. It does not estimate customer welfare or identify one update rule as uniformly preferable. Within this scope, the results establish that representations, fitting policies, and environmental responses require joint evaluation under graph feedback.

7 AI Use Statement

Gen AI tools were used to provide feedback on the paper, i.e., the presentation and reader friendliness, and seeking revisions to certain phrases/sentences/paragraphs. The authors reviewed all AI-assisted revisions, and if needed modified the revisions. The authors take responsibility for the final content of this work.

REFERENCES

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

Gavin Brown, Shlomi Hod, and Iden Kalemaj. Performative prediction in a stateful world. In *International conference on artificial intelligence and statistics*, pp. 6045–6061. PMLR, 2022.

Tiffany Cai, Hongseok Namkoong, and Steve Yadlowsky. Diagnosing model performance under distribution shift, 2026.

Itay Eilat and Nir Rosenfeld. Performative recommendation: diversifying content via strategic incentives. In *International Conference on Machine Learning*, pp. 9082–9103. PMLR, 2023.

Itay Eilat, Ben Finkelshtein, Chaim Baskin, and Nir Rosenfeld. Strategic classification with graph neural networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TuHkVOjSAR.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking LLMs as semantic encoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3251–3265, 2026. doi: 10.18653/v1/2026.acl-long.147.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Gunnar König, Hidde Fokkema, Timo Freiesleben, Celestine Mendler-Dünner, and Ulrike Luxburg. Performative validity of recourse explanations. In *Advances in Neural Information Processing Systems*, volume 38, pp. 139334–139370, 2025. doi: 10.52202/085713-4652.

Sean P. Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, 2nd edition, 2009. doi: 10.1017/CBO9780511626630.

John P Miller, Juan C Perdomo, and Tijana Zrnic. Outside the echo chamber: Optimizing the performative risk. In *International Conference on Machine Learning*, pp. 7710–7720. PMLR, 2021.

Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International conference on machine learning*, pp. 7599–7609. PMLR, 2020.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6861–6871. PMLR, 09–15 Jun 2019.

A NOTATION AND TERMINOLOGY

Table 4 summarizes notation used across the theoretical framework and empirical studies. Symbols used only within individual proofs are defined locally and omitted here. We use *outcome* for an evolving binary variable, *label* for an observed value used in fitting or evaluation (including fixed historical labels in the application), and *environmental response* for the mechanism that maps a deployed prediction and current state to a subsequent state.

Table 4: Core notation used throughout the paper.

Symbol	Definition
Environment and stochastic process	
$\mathcal{I}$	**Prediction units.** Objects for which the classifier produces individual predictions: nodes in the synthetic study and specified user–product pairs in the application.
$\mathcal{G}_t$	**Attributed graph.** Nodes, edges, and response-relevant attributes at round t, including $\boldsymbol{X}_t$. The synthetic graph is $(V, E, \boldsymbol{X}_t)$; the application also retains current description choices needed for edit costs.
$\boldsymbol{y}_t$	**Outcome vector.** Binary outcomes associated with the prediction units at round t; observed coordinates serve as training or evaluation labels.
$s_t = (\mathcal{G}_t, \boldsymbol{y}_t)$	**Environment state.** One realized configuration of the graph, features, and outcomes.
$\mathcal{S}$	**State space.** Set of possible environment states.
$\mathcal{H} = \mathcal{C} \times \{0,1\}^n$	**Synthetic state space.** Compact state space of mutable-feature vectors $\boldsymbol{x} \in \mathcal{C}$ and binary-outcome vectors $\boldsymbol{y} \in \{0,1\}^n$ used in the invariant-distribution analysis.
μ_t	**State distribution.** Probability distribution over possible environment states at round t.
$P(\cdot \mid s, \boldsymbol{u})$	**Environmental transition.** Distribution of the next state given current state s and deployment signal or action $\boldsymbol{u}$.
$\mathsf{K}_{\boldsymbol{\theta}}(\cdot \mid s)$	**Deployed transition kernel.** Next-state distribution obtained by deploying classifier parameter $\boldsymbol{\theta}$ in state s.
$\Phi_{\boldsymbol{\theta}}(\mu)$	**Distribution update.** State distribution after one deployment of a fixed classifier with parameter $\boldsymbol{\theta}$ from initial state distribution μ.
Graph prediction and fitting	
$\boldsymbol{A}, \boldsymbol{S}, \mathbf{N}$	**Graph operators.** Adjacency matrix, normalized propagation matrix with self-loops, and row-normalized neighbor matrix without self-loops, respectively.
L, U	**Training and test sets.** Fixed index sets used for fitting and held-out evaluation.
$\mathbf{Z}_{\mathrm{R}}, \mathbf{Z}_{\mathrm{G}}, \mathbf{Z}_{\mathrm{C}}$	**Representations.** Raw, graph-propagated, and concatenated raw-plus-propagated feature representations.
$f_{\boldsymbol{\theta}}$	**Classifier.** Prediction rule with parameter $\boldsymbol{\theta}$; the primary study uses $\boldsymbol{\theta} = \boldsymbol{w}$.
h_i, p_i	**Logit and probability.** Classifier logit and predicted probability for prediction unit i.
$\mathcal{L}(\boldsymbol{X}, \boldsymbol{y}_L; \boldsymbol{w})$	**Training objective.** Mean binary cross-entropy on L plus quadratic regularization.
$\mathcal{T}(O(s))$	**Fitting map.** Classifier fitted to the current observation. In the synthetic instance, $\mathcal{T}(\boldsymbol{X}, \boldsymbol{y}_L)$ is the unique minimizer of the regularized objective.
$\mathcal{R}(f, s)$	**Held-out predictive risk.** Mean unpenalized loss of classifier f on the specified evaluation units in environment state s.
$r(s, \boldsymbol{w})$	**Coefficient-based risk.** Expression for $\mathcal{R}(f_{\boldsymbol{w}}, s)$ used in the primary model.

Table 4 continued.

Symbol	Definition
Environmental dynamics	
$x_{i,t}, b_i$	**Mutable and reference features.** Current mutable feature and its initial reference value for node i.
a, c, χ	**Feature-response parameters.** Adjustment rate, response magnitude, and response direction.
$q_{i,t}, \rho_{i,t}, \varepsilon$	**Outcome-update quantities.** Conditional probability of outcome 1 when resampled, final Bernoulli probability after persistence, and outcome-resampling probability, respectively.
$\mathcal{F}_X, Q$	**State-update components.** Deterministic feature-update map and stochastic outcome kernel.
Policies and performance comparisons	
$\pi \in \{\text{refit}, \text{fixed}\}$	**Classifier-update policy.** Rule that either refits after each transition or retains the initial classifier.
$s_t^\pi, \boldsymbol{w}_t^\pi$	**Policy-indexed state and classifier.** Environment state and classifier parameter at round t under policy π; their sequences form the policy-induced trajectory.
$J_{t_0,H}(\pi)$	**Finite-horizon policy loss.** Expected held-out loss averaged along the trajectory induced by π over the stated evaluation window.
$\Delta_{t_0,H}$	**Policy loss difference.** Refitting-policy loss minus fixed-classifier-policy loss.
Recourse application	
A_i, τ	**Audience and threshold.** Fixed user audience for product i and the eligibility threshold.
$\mathbf{X}^{i,k}$	**Candidate feature matrix.** Product-feature matrix when product i uses description candidate k.
$\boldsymbol{\psi}_{ui}(\mathbf{X}^{i,k})$	**Pair design vector.** Features used to score user–product pair (u, i) under candidate k.

B Complete Environment Specifications

B.1 Synthetic node classification

Let $\mathcal{G}_t = (V, E, \boldsymbol{X}_t)$, where the node set V and edge set E remain fixed across rounds, and let $n \coloneqq |V|$. Each node $i \in V$ has an immutable feature $r_i \in \mathbb{R}$, a mutable feature $x_{i,t} \in \mathbb{R}$, and a binary outcome $y_{i,t} \in \{0, 1\}$. Writing $\mathbf{r} = (r_i)_{i \in V}$, $\boldsymbol{x}_t = (x_{i,t})_{i \in V}$, and $\boldsymbol{y}_t = (y_{i,t})_{i \in V}$, define

$$\boldsymbol{X}_t \coloneqq [\, \mathbf{r} \mid \boldsymbol{x}_t \,] \in \mathbb{R}^{n \times 2}.$$

The immutable vector $\mathbf{r}$ remains fixed, and the prediction task is to classify each node's current outcome.

We use a fixed transductive split of the nodes into training, validation, and test sets. The learner observes the complete graph and all node features. Only the current outcomes on the training set L are observed for fitting, and these values form the training-label vector $\boldsymbol{y}_{L,t}$. Fitting uses only these training outcomes; the outcomes on the fixed test set U are reserved for held-out evaluation. The environment maintains outcomes for all nodes. Since V, E, and $\mathbf{r}$ remain fixed, its evolving state is fully represented by

$$s_t = (\boldsymbol{x}_t, \boldsymbol{y}_t) \in \mathcal{S} \coloneqq \mathbb{R}^n \times \{0, 1\}^n.$$

Environmental change affects mutable features and outcomes, while graph membership and topology remain fixed.

Let $\boldsymbol{A}$ be fixed undirected adjacency with zero diagonal. For prediction, we use the standard normalization (Kipf & Welling, 2017)

$$\widetilde{\boldsymbol{A}} \coloneqq \boldsymbol{A} + \boldsymbol{I}, \qquad \widetilde{D}_{ii} \coloneqq \sum_j \widetilde{A}_{ij}, \qquad \boldsymbol{S} \coloneqq \widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1/2}. \tag{13}$$

Let $N_{ij} \coloneqq A_{ij}/\sum_k A_{ik}$ for a nonisolated node and $N_{ij} \coloneqq 0$ for an isolated node. Outcome transitions use $\mathbf{N}$, excluding self; own features enter separately.

Representations and predictions. We compare three fixed representation maps, with a common graph and initial preprocessing:

$$\begin{aligned} \mathbf{Z}_{\mathrm{R}}(\boldsymbol{X}) &\coloneqq [\,\boldsymbol{X} \mid \mathbf{1}\,], \\ \mathbf{Z}_{\mathrm{G}}(\boldsymbol{X}) &\coloneqq [\,\boldsymbol{S}^K\boldsymbol{X}\mathbf{D}_K \mid \mathbf{1}\,], \\ \mathbf{Z}_{\mathrm{C}}(\boldsymbol{X}) &\coloneqq [\,\boldsymbol{X} \mid \boldsymbol{S}^K\boldsymbol{X}\mathbf{D}_K \mid \mathbf{1}\,]. \end{aligned} \tag{14}$$

R retains the original features, G aggregates features over K graph steps, and C retains both. The fixed diagonal matrix $\mathbf{D}_K$ scales the propagated channels; Appendix E.2 specifies its construction and the settings used in the experiments.

Within a specified representation, write $\mathbf{Z}(\boldsymbol{X}) \in \mathbb{R}^{n\times d}$, with row vector $\boldsymbol{z}_i(\boldsymbol{X})^\top$. For coefficients $\boldsymbol{w} \in \mathbb{R}^d$, define

$$h_i(\boldsymbol{w};\boldsymbol{X}) \coloneqq \boldsymbol{z}_i(\boldsymbol{X})^\top\boldsymbol{w}, \qquad p_i(\boldsymbol{w};\boldsymbol{X}) \coloneqq \sigma(h_i(\boldsymbol{w};\boldsymbol{X})), \qquad \sigma(z) \coloneqq (1+e^{-z})^{-1}. \tag{15}$$

The vector $\boldsymbol{p}(\boldsymbol{w};\boldsymbol{X}) \coloneqq (p_1(\boldsymbol{w};\boldsymbol{X}),\dots,p_n(\boldsymbol{w};\boldsymbol{X}))^\top$ collects the predicted probabilities. Only $\boldsymbol{w}$ is learned; the representation is recomputed from current features at each round using the same graph operator and scales.

Training and refitting. Let $m \coloneqq |L| > 0$ be the number of training nodes, and let $\boldsymbol{y}_L$ collect their current observed labels, with y_i denoting the label of node $i \in L$. We fit $\boldsymbol{w}$ by minimizing mean binary cross-entropy plus a quadratic penalty:

$$\mathcal{L}(\boldsymbol{X},\boldsymbol{y}_L;\boldsymbol{w}) \coloneqq \frac{1}{m}\sum_{i\in L}\left[\log\left(1+e^{\boldsymbol{z}_i(\boldsymbol{X})^\top\boldsymbol{w}}\right) - y_i\boldsymbol{z}_i(\boldsymbol{X})^\top\boldsymbol{w}\right] + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2. \tag{16}$$

Here $\lambda > 0$. Define the fitting map

$$\mathcal{T}(\boldsymbol{X},\boldsymbol{y}_L) \coloneqq \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{R}^d} \mathcal{L}(\boldsymbol{X},\boldsymbol{y}_L;\boldsymbol{w}). \tag{17}$$

Since ridge regularization makes $\mathcal{L}(\boldsymbol{X},\boldsymbol{y}_L;\cdot)$ strongly convex, the fit is unique. At round t, refitting yields $\boldsymbol{w}_t = \mathcal{T}(\boldsymbol{X}_t,\boldsymbol{y}_{L,t})$ from the current snapshot; each fit uses only observations from that round. Deployment can then change the features and outcomes available at the next round.

Held-out predictive loss. For a fixed nonempty test set $U \subseteq V \setminus L$, define

$$\ell(z,y) \coloneqq \log(1+e^z) - yz, \qquad r(s,\boldsymbol{w}) \coloneqq \frac{1}{|U|}\sum_{i\in U}\ell(h_i(\boldsymbol{w};\boldsymbol{X}),y_i) = \mathcal{R}(f_{\boldsymbol{w}},s). \tag{18}$$

The test outcomes enter evaluation only. The learner minimizes the regularized training objective $\mathcal{L}$ in Equation (16), which is distinct from r.

Deployment-induced feature adjustment. Let $b_i \coloneqq x_{i,0}$ denote node i's reference feature after initial standardization. At round t, the deployed classifier produces probabilities $\boldsymbol{p}_t = \boldsymbol{p}(\boldsymbol{w}_t;\boldsymbol{X}_t)$. In this synthetic study, the deployment signal is the probability vector itself, $\boldsymbol{u}_t = \boldsymbol{p}_t$. We model the resulting environmental response by

$$x_{i,t+1} = (1-a)x_{i,t} + a[b_i + \chi c(2p_{i,t}-1)], \qquad a \in [0,1], \quad c > 0, \quad \chi \in \{-1,+1\}. \tag{19}$$

This rule specifies how the current mutable feature adjusts toward a prediction-dependent target. The immutable feature is unchanged. For $0 < a < 1$, a node retains part of its current feature while moving toward the prediction-dependent target. The reference provides a restoring tendency. Under $\chi = +1$, high scores raise that target; the comparison $\chi = -1$ reverses this direction. At $a = 0$, no feature adjustment occurs.

Outcome evolution. Features are updated simultaneously before new outcomes are sampled. Define $\Delta x_{j,t} \coloneqq x_{j,t+1} - x_{j,t}$ and let $\mathcal{N}(i)$ be the original neighbor set. The probability of state 1 when an outcome is resampled is

$$q_{i,t} \coloneqq \sigma\left(x_{i,t+1} + \mathrm{mean}_{j\in\mathcal{N}(i)}\, x_{j,t+1} + \mathrm{mean}_{j\in\mathcal{N}(i)}\, \Delta x_{j,t}\right). \tag{20}$$

Empty means are zero. Own conditions, neighboring conditions, and signed neighboring changes all favor state 1 when they increase. The unit coefficients are fixed parameters of the synthetic environment. Opposing neighboring changes can cancel in the last term.

To allow persistence of the previous state, set

$$\rho_{i,t} \coloneqq (1-\varepsilon)y_{i,t} + \varepsilon q_{i,t}, \qquad y_{i,t+1} \sim \mathrm{Bernoulli}(\rho_{i,t}), \qquad \varepsilon \in [0,1]. \tag{21}$$

Equivalently, retain the outcome with probability $1-\varepsilon$ and otherwise draw it with probability $q_{i,t}$ of state 1. The transition probabilities from 0 to 1 and from 1 to 0 are respectively $\varepsilon q_{i,t}$ and $\varepsilon(1-q_{i,t})$. Conditional on the current state and updated features, the node draws are independent. They remain coupled unconditionally through the graph and the evolving features. Setting $\varepsilon = 0$ preserves every outcome exactly; random outcome changes remain possible when only $a = 0$.

Repeated deployment. Initialize $\boldsymbol{w}_0 = \mathcal{T}(\boldsymbol{X}_0, \boldsymbol{y}_{L,0})$. One feature adjustment and one outcome update precede each refit:

$$\boldsymbol{X}_{t+1} = \mathcal{F}_X(\boldsymbol{X}_t, \boldsymbol{w}_t), \qquad \boldsymbol{y}_{t+1} \sim Q(\cdot \mid s_t, \boldsymbol{X}_{t+1}), \qquad \boldsymbol{w}_{t+1} = \mathcal{T}(\boldsymbol{X}_{t+1}, \boldsymbol{y}_{L,t+1}). \tag{22}$$

The maps $\mathcal{F}_X$ and Q are defined by Equations (19)–(21), and the updated state is retained for the next deployment. Since outcomes are updated stochastically, the process is stochastic. For G and C, the graph enters the predictions through feature propagation; for every representation, it enters the response through neighbor-dependent outcome transitions. Appendix E.2 specifies the explicit deployed transition kernel.

B.2 Explanation recourse on a user–product graph

We instantiate the graph-feedback framework as repeated explanation recourse on a fixed user–product graph. A classifier assigns each specified user–product pair a probability of a favorable rating. For each product, the platform aggregates these predictions over a fixed audience and, when the resulting score is below an eligibility threshold, recommends a fact-preserving description edit intended to cross that threshold. Implementing an edit changes the features used in subsequent predictions, after which the platform may refit the classifier and issue another recourse recommendation.

The prediction units are specified user–product pairs (u, i), and each binary label records whether the pair's historical rating is favorable. Fixed training interactions form the bipartite graph. Future purchases, reviews, and rating outcomes lie outside the simulation: the interaction edges and historical rating labels remain fixed. Feedback operates through changes to product descriptions and the pair representations derived from them.

At round t, the rows of $\boldsymbol{X}_t$ contain the current product-description embeddings, and the state s_t also records the current description choices needed to evaluate later edit costs. Training review summaries, the text encoder, graph operators, feature scales, audiences, and candidate sets remain fixed. Graph edges remain fixed while graph-derived pair features change with $\boldsymbol{X}_t$. Appendix G specifies the data split, text representation, and information restrictions.

Prediction and refitting. For representation $j \in \{\mathrm{R}, \mathrm{G}, \mathrm{C}\}$, the pair design $\boldsymbol{\psi}^j_{ui}(\boldsymbol{X})$ combines fixed review summaries with the current product and graph-derived features. R uses raw product features, G uses propagated product features, and C uses both; all three also include user-history means computed from the interaction graph. Given readout coefficients $\mathbf{v}$, the predicted probability for pair (u, i) is

$$p^j_{ui}(\mathbf{v}; \boldsymbol{X}) \coloneqq \sigma\Big(\boldsymbol{\psi}^j_{ui}(\boldsymbol{X})^\top \mathbf{v}\Big). \tag{23}$$

Let L be the fixed set of training pairs and let y_{ui} denote their historical rating labels. Since these labels remain fixed, refitting responds to the changed pair representations by updating only the

readout coefficients:

$$\mathcal{T}_j(O(s)) \coloneqq \arg\min_{\mathbf{v}} \left\{ \frac{1}{|L|} \sum_{(u,i)\in L} \ell\Big(\boldsymbol{\psi}_{ui}^{j}(\boldsymbol{X})^{\top}\mathbf{v}, y_{ui}\Big) + \frac{\lambda}{2}\|\mathbf{v}\|_2^2 \right\}, \tag{24}$$

Here ℓ is the logistic loss in Equation (18). Since $\lambda > 0$ and every coefficient is regularized, the readout fit is unique. The refitting policy therefore uses $\mathbf{v}_t = \mathcal{T}_j(O(s_t))$; the fixed-classifier policy retains the initial coefficients $\mathbf{v}_0$. Appendix G specifies the complete pair designs and fitting objective.

Recourse recommendations and state transition. Predictions determine which description edits are implemented. Each product i has a fixed audience A_i and a finite set of reviewed, fact-preserving candidates. For candidate description k, let $\boldsymbol{X}^{i,k}$ replace only product i's current description embedding. The candidate's audience score under coefficients $\mathbf{v}_t$ is

$$g_i^j(k; \mathbf{v}_t) \coloneqq \frac{1}{|A_i|} \sum_{u\in A_i} p_{ui}^{j}(\mathbf{v}_t; \boldsymbol{X}^{i,k}). \tag{25}$$

A product whose current score is below τ selects the least-cost permitted candidate with score at least τ. A product already above the threshold, or without a qualifying candidate, remains unchanged. The selected candidate is the recourse recommendation. The corresponding description edits are implemented simultaneously, after which the graph-derived features are recomputed. Consequently, changing one product description can alter pair representations elsewhere through graph propagation.

Let $\Gamma_{\mathbf{v}}(s)$ denote the state obtained after applying these edits, including the updated product features and description choices. For fixed coefficients $\mathbf{v}$, candidate sets, and tie-breaking, this mapping is deterministic. The deployed transition kernel is therefore the point mass

$$\mathsf{K}_{\mathbf{v}}(C \mid s) \coloneqq \mathbf{1}\{\Gamma_{\mathbf{v}}(s) \in C\}. \tag{26}$$

Thus the next state is exactly $\Gamma_{\mathbf{v}}(s)$. Historical rating labels remain fixed throughout this transition.

Repeated deployment and evaluation. Both policies start from the common initial fit $\mathbf{v}_0 = \mathcal{T}_j(O(s_0))$. Their repeated updates are

$$s_{t+1}^{\pi} = \Gamma_{\mathbf{v}_t^{\pi}}(s_t^{\pi}), \qquad \mathbf{v}_{t+1}^{\pi} = \begin{cases} \mathcal{T}_j(O(s_{t+1}^{\pi})), & \pi = \text{refit}, \\ \mathbf{v}_0, & \pi = \text{fixed}. \end{cases} \tag{27}$$

This recursion specializes Equation (2) to the deterministic recourse transition. Under refitting, an updated description can affect both the next pair representations and the next coefficients; under the fixed-classifier policy, only the representations change. The simulation assumes that every recourse recommendation is implemented as a description edit and generates no post-edit customer ratings.

The held-out risk $\mathcal{R}(f, s)$ is the mean logistic loss on a fixed set of historical user–product pairs. Evaluating the same pairs along each policy-induced trajectory measures predictive loss as product representations change. Separately, post-refit threshold retention records whether a product with an implemented description edit remains above the eligibility threshold after refitting. Section 4.3 reports both quantities. Appendix G specifies the candidate screening, calibration, and cohort definitions.

Having specified both graph-feedback processes, we first analyze the stochastic node process and then compare predictive performance under the assumptions of each instance.

C Proofs for the Persistent Classification Model

All claims in this section condition on a fixed realized dataset, finite initial features, fixed graph and split, and fixed response parameters. The bounds may depend on the initial data.

C.1 Proof of Proposition 1

The three parts below correspond to the three statements in Proposition 1.

Part 1: The mutable features remain in $\mathcal{C}$. For every node i and round t, the predicted probability satisfies $p_{i,t} \in [0,1]$. Hence $2p_{i,t}-1 \in [-1,1]$, and the prediction-dependent target in Equation (19) obeys

$$b_i + \chi c(2p_{i,t} - 1) \in [b_i - c, b_i + c]$$

for either $\chi = +1$ or $\chi = -1$. Since $a \in [0,1]$, $x_{i,t} \in [b_i - c, b_i + c]$ implies that $x_{i,t+1}$ is a convex combination of $x_{i,t}$ and another point in the same interval. Hence $x_{i,t+1} \in [b_i - c, b_i + c]$. Since $x_{i,0} = b_i \in [b_i - c, b_i + c]$, induction over t yields $x_{i,t} \in [b_i - c, b_i + c]$ for every node and every round $t \geq 0$. Equivalently, $\boldsymbol{x}_t \in \mathcal{C}$.

Part 2: Every refitting problem has exactly one solution. Fix a finite feature matrix $\boldsymbol{X}$ and binary training labels $\boldsymbol{y}_L$. Let $\mathbf{Z}_L = \mathbf{Z}(\boldsymbol{X})_L$ be the representation matrix restricted to the $m = |L|$ training nodes. During this refit, $\boldsymbol{X}$ and hence $\mathbf{Z}_L$ are fixed; only $\boldsymbol{w}$ is optimized. Writing $\boldsymbol{p}_L$ for the vector of predicted probabilities on the training nodes, the gradient is

$$\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{X}, \boldsymbol{y}_L; \boldsymbol{w}) = \frac{1}{m} \mathbf{Z}_L^\top (\boldsymbol{p}_L - \boldsymbol{y}_L) + \lambda \boldsymbol{w}.$$

Differentiating once more with respect to $\boldsymbol{w}$ yields

$$\nabla^2_{\boldsymbol{w}\boldsymbol{w}} \mathcal{L}(\boldsymbol{X}, \boldsymbol{y}_L; \boldsymbol{w}) = \frac{1}{m} \mathbf{Z}_L^\top \boldsymbol{\Omega}(\boldsymbol{w}) \mathbf{Z}_L + \lambda \boldsymbol{I} \succeq \lambda \boldsymbol{I}, \tag{28}$$

where $\boldsymbol{\Omega}(\boldsymbol{w})$ is diagonal with entries $p_i(\boldsymbol{w}; \boldsymbol{X})(1 - p_i(\boldsymbol{w}; \boldsymbol{X})) \in [0, 1/4]$. To see the matrix inequality, for any direction $\boldsymbol{u} \in \mathbb{R}^d$,

$$\boldsymbol{u}^\top \left(\frac{1}{m} \mathbf{Z}_L^\top \boldsymbol{\Omega} \mathbf{Z}_L \right) \boldsymbol{u} = \frac{1}{m} \sum_{i \in L} p_i(1 - p_i)(\boldsymbol{z}_i^\top \boldsymbol{u})^2 \geq 0.$$

Thus the logistic-loss term has nonnegative curvature, while the quadratic penalty contributes curvature λ in every coefficient direction. The complete objective is therefore λ-strongly convex, which can have at most one minimizer.

It remains to show that a minimizer exists. Binary cross-entropy is nonnegative, and hence

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{y}_L; \boldsymbol{w}) \geq \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 .$$

Since $\lambda > 0$, the right-hand side tends to infinity as $\|\boldsymbol{w}\|_2 \to \infty$, and hence the objective is coercive. Since it is also continuous on $\mathbb{R}^d$, it attains a minimizer by Weierstrass' theorem (Bertsekas, 1999, Proposition A.8). Strong convexity then makes the minimizer unique, including when all observed training labels belong to the same class. This last conclusion uses the assumption that every coefficient, including the intercept, is regularized.

Part 3: The fitted coefficients are uniformly bounded. Let $\widehat{\boldsymbol{w}} = \mathcal{T}(\boldsymbol{X}, \boldsymbol{y}_L)$ denote the unique minimizer. Its objective value cannot exceed the value at the zero vector. Since every logit and the quadratic penalty are zero at $\boldsymbol{w} = \mathbf{0}$,

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{y}_L; \mathbf{0}) = \frac{1}{m} \sum_{i \in L} \left[\log(1 + e^0) - y_i \, 0 \right] = \log 2.$$

Since the mean binary cross-entropy term is nonnegative,

$$\frac{\lambda}{2} \|\widehat{\boldsymbol{w}}\|_2^2 \leq \mathcal{L}(\boldsymbol{X}, \boldsymbol{y}_L; \widehat{\boldsymbol{w}}) \leq \mathcal{L}(\boldsymbol{X}, \boldsymbol{y}_L; \mathbf{0}) = \log 2.$$

Rearranging yields $\|\widehat{\boldsymbol{w}}\|_2 \leq \sqrt{2 \log 2 / \lambda} = B_w$. This bound is independent of the particular feature matrix and label vector. □

C.2 Continuity of Exact Refitting and Proof of Proposition 2

We first establish the continuity property needed for the transition operator.

Lemma 1 (Continuity of exact refitting)**.** *Under the assumptions of Proposition 1, suppose that the fixed representation map $\boldsymbol{X} \mapsto \mathbf{Z}(\boldsymbol{X})$ is continuous. For each fixed training-label vector $\boldsymbol{y}_L$, the map*

$$\boldsymbol{x} \longmapsto \mathcal{T}([\,\mathbf{r} \mid \boldsymbol{x}\,], \boldsymbol{y}_L)$$

is continuous on $\mathcal{C}$.

Proof. Fix $\boldsymbol{y}_L$ and let $\boldsymbol{x}_k \to \boldsymbol{x}$ in $\mathcal{C}$. Define $\boldsymbol{X}_k \coloneqq [\,\mathbf{r} \mid \boldsymbol{x}_k\,]$ and $\boldsymbol{X} \coloneqq [\,\mathbf{r} \mid \boldsymbol{x}\,]$. Let $\widehat{\boldsymbol{w}}_k \coloneqq \mathcal{T}(\boldsymbol{X}_k, \boldsymbol{y}_L)$ minimize the objective at $\boldsymbol{x}_k$, and let $\widehat{\boldsymbol{w}} \coloneqq \mathcal{T}(\boldsymbol{X}, \boldsymbol{y}_L)$ minimize the objective at $\boldsymbol{x}$. Part 3 of Proposition 1 implies that $\widehat{\boldsymbol{w}}_k \in \mathcal{W}$ for every k and that $\widehat{\boldsymbol{w}} \in \mathcal{W}$, where

$$\mathcal{W} \coloneqq \{\boldsymbol{w} : \|\boldsymbol{w}\|_2 \le B_w\}.$$

Step 1: Uniform convergence of the training objectives. Define

$$G(\boldsymbol{x}, \boldsymbol{w}) \coloneqq \mathcal{L}([\,\mathbf{r} \mid \boldsymbol{x}\,], \boldsymbol{y}_L; \boldsymbol{w}), \qquad F_k(\boldsymbol{w}) \coloneqq G(\boldsymbol{x}_k, \boldsymbol{w}), \quad F(\boldsymbol{w}) \coloneqq G(\boldsymbol{x}, \boldsymbol{w}).$$

Since G is continuous and $\mathcal{C} \times \mathcal{W}$ is compact, the Heine–Cantor theorem implies that G is uniformly continuous. For each k, define the objective approximation error and its largest magnitude over $\mathcal{W}$ by

$$e_k(\boldsymbol{w}) \coloneqq F_k(\boldsymbol{w}) - F(\boldsymbol{w}), \qquad \delta_k \coloneqq \sup_{\boldsymbol{w} \in \mathcal{W}} |e_k(\boldsymbol{w})|.$$

Uniform continuity and $\boldsymbol{x}_k \to \boldsymbol{x}$ imply $\delta_k \to 0$.

Step 2: Convergence of the fitted coefficients. For every $\boldsymbol{w}$, strong convexity yields

$$F_k(\boldsymbol{w}) \ge F_k(\widehat{\boldsymbol{w}}_k) + \nabla F_k(\widehat{\boldsymbol{w}}_k)^\top (\boldsymbol{w} - \widehat{\boldsymbol{w}}_k) + \frac{\lambda}{2} \|\boldsymbol{w} - \widehat{\boldsymbol{w}}_k\|_2^2 .$$

Since $\widehat{\boldsymbol{w}}_k$ is the unconstrained minimizer of F_k, $\nabla F_k(\widehat{\boldsymbol{w}}_k) = \mathbf{0}$. Taking $\boldsymbol{w} = \widehat{\boldsymbol{w}}$ and rearranging yields

$$\frac{\lambda}{2} \|\widehat{\boldsymbol{w}}_k - \widehat{\boldsymbol{w}}\|_2^2 \le F_k(\widehat{\boldsymbol{w}}) - F_k(\widehat{\boldsymbol{w}}_k). \tag{29}$$

Since $\widehat{\boldsymbol{w}}_k, \widehat{\boldsymbol{w}} \in \mathcal{W}$, the definition of δ_k yields $|e_k(\widehat{\boldsymbol{w}})| \le \delta_k$ and $|e_k(\widehat{\boldsymbol{w}}_k)| \le \delta_k$.

Moreover, since $\widehat{\boldsymbol{w}}$ minimizes F, $F(\widehat{\boldsymbol{w}}) - F(\widehat{\boldsymbol{w}}_k) \le 0$. It follows that

$$\begin{aligned} F_k(\widehat{\boldsymbol{w}}) - F_k(\widehat{\boldsymbol{w}}_k) &= F(\widehat{\boldsymbol{w}}) - F(\widehat{\boldsymbol{w}}_k) + e_k(\widehat{\boldsymbol{w}}) - e_k(\widehat{\boldsymbol{w}}_k) \\ &\le F(\widehat{\boldsymbol{w}}) - F(\widehat{\boldsymbol{w}}_k) + |e_k(\widehat{\boldsymbol{w}})| + |e_k(\widehat{\boldsymbol{w}}_k)| \\ &\le F(\widehat{\boldsymbol{w}}) - F(\widehat{\boldsymbol{w}}_k) + 2\delta_k \\ &\le 2\delta_k. \end{aligned} \tag{30}$$

Combining Equations (29) and (30) yields

$$\|\widehat{\boldsymbol{w}}_k - \widehat{\boldsymbol{w}}\|_2^2 \le \frac{4\delta_k}{\lambda} \longrightarrow 0.$$

Since the convergent sequence $\boldsymbol{x}_k \to \boldsymbol{x}$ was arbitrary, the exact-refitting map is continuous for the fixed training-label vector $\boldsymbol{y}_L$. □

Proof of Proposition 2. Recall that $\mathcal{H} \coloneqq \mathcal{C} \times \{0,1\}^n$ is the compact state space of mutable-feature and binary-outcome vectors. The proof follows the effect of one deployment on distributions over graph states. We first show that the transition preserves bounded continuity, then take a weak limit of averaged state distributions, and finally show that this limit is invariant.

Step 1: Show that one transition preserves bounded continuity. Fix $\boldsymbol{y} \in \{0,1\}^n$ and consider states $s = (\boldsymbol{x}, \boldsymbol{y})$ with $\boldsymbol{x} \in \mathcal{C}$.

First, define the current feature matrix by

$$\boldsymbol{X}(s) \coloneqq [\,\mathbf{r} \mid \boldsymbol{x}\,].$$

Since $\mathbf{r}$ is fixed, the map $\boldsymbol{x} \mapsto \boldsymbol{X}(s)$ is continuous.

The exactly refitted coefficients are

$$\widehat{\boldsymbol{w}}(s) \coloneqq \mathcal{T}(\boldsymbol{X}(s), \boldsymbol{y}_L).$$

The training-label vector $\boldsymbol{y}_L$ is fixed when $\boldsymbol{y}$ is fixed. Lemma 1 therefore implies that

$$\boldsymbol{x} \longmapsto \widehat{\boldsymbol{w}}(s)$$

is continuous.

By assumption, the fixed map $\boldsymbol{X} \mapsto \mathbf{Z}(\boldsymbol{X})$ is continuous. Since matrix multiplication and the coordinatewise sigmoid are continuous, the prediction map

$$(\boldsymbol{X}, \boldsymbol{w}) \longmapsto \boldsymbol{p}(\boldsymbol{w}; \boldsymbol{X}) = \sigma(\mathbf{Z}(\boldsymbol{X})\boldsymbol{w})$$

is jointly continuous in $\boldsymbol{X}$ and $\boldsymbol{w}$. Since $\boldsymbol{x} \mapsto \boldsymbol{X}(s)$ and $\boldsymbol{x} \mapsto \widehat{\boldsymbol{w}}(s)$ are continuous, it follows that

$$\boldsymbol{x} \longmapsto \boldsymbol{p}(\widehat{\boldsymbol{w}}(s); \boldsymbol{X}(s))$$

is continuous.

The next mutable-feature vector is

$$\mathbf{f}(s) \coloneqq (1-a)\boldsymbol{x} + a\left[\boldsymbol{b} + \chi c\left(2\boldsymbol{p}(\widehat{\boldsymbol{w}}(s); \boldsymbol{X}(s)) - \mathbf{1}\right)\right].$$

This expression consists of sums and scalar multiples of continuous functions of $\boldsymbol{x}$. Therefore,

$$\boldsymbol{x} \longmapsto \mathbf{f}(s)$$

is continuous.

Next, write the conditional probability of outcome 1 under resampling from Equation (20) as

$$q_i(s) \coloneqq \sigma\left(f_i(s) + \mathrm{mean}_{j\in\mathcal{N}(i)}\, f_j(s) + \mathrm{mean}_{j\in\mathcal{N}(i)}\big(f_j(s) - x_j\big)\right),$$

where an empty mean is zero. Since each term inside the sigmoid is continuous in $\boldsymbol{x}$, $q_i(s)$ is continuous in $\boldsymbol{x}$.

By Equation (21), the probability that node i has outcome 1 in the next round is

$$\rho_i(s) \coloneqq (1-\varepsilon)y_i + \varepsilon q_i(s).$$

Since y_i is fixed on the current component and $q_i(s)$ is continuous in $\boldsymbol{x}$, $\rho_i(s)$ is continuous in $\boldsymbol{x}$.

For a possible next outcome vector $\boldsymbol{y}' \in \{0,1\}^n$, define

$$\pi_{\boldsymbol{y}'}(s) \coloneqq \Pr(\boldsymbol{y}_{t+1} = \boldsymbol{y}' \mid s_t = s).$$

Conditional independence of the node-level outcome updates yields

$$\pi_{\boldsymbol{y}'}(s) = \prod_{i=1}^{n}\left[y_i'\rho_i(s) + (1-y_i')\big(1-\rho_i(s)\big)\right].$$

Each factor is continuous in $\boldsymbol{x}$, and a finite product of continuous functions is continuous. Therefore,

$$\boldsymbol{x} \longmapsto \pi_{\boldsymbol{y}'}(s)$$

is continuous.

Now let $\varphi : \mathcal{H} \to \mathbb{R}$ be bounded and continuous. Applying the transition operator P to φ yields

$$(\mathsf{P}\varphi)(s) \coloneqq \mathbb{E}[\varphi(s_{t+1}) \mid s_t = s] = \sum_{\boldsymbol{y}'\in\{0,1\}^n} \pi_{\boldsymbol{y}'}(s)\,\varphi(\mathbf{f}(s), \boldsymbol{y}').$$

For each fixed $\boldsymbol{y}'$, continuity of $\mathbf{f}$ and φ implies that

$$\boldsymbol{x} \longmapsto \varphi(\mathbf{f}(s), \boldsymbol{y}')$$

is continuous. We have also shown that $\boldsymbol{x} \mapsto \pi_{\boldsymbol{y}'}(s)$ is continuous. Hence, every summand in the display above is continuous in $\boldsymbol{x}$. Since the sum contains only finitely many terms, $(\mathsf{P}\varphi)(s)$ is continuous in $\boldsymbol{x}$ for the fixed current outcome vector $\boldsymbol{y}$.

The argument holds for every $\boldsymbol{y} \in \{0,1\}^n$. Since $\{0,1\}^n$ has the discrete topology, $\mathsf{P}\varphi$ is continuous on the full state space $\mathcal{H}$.

Finally, the weights $\pi_{\boldsymbol{y}'}(s)$ are nonnegative and sum to one. Therefore,

$$|(\mathsf{P}\varphi)(s)| \leq \sum_{\boldsymbol{y}' \in \{0,1\}^n} \pi_{\boldsymbol{y}'}(s)\, |\varphi(\mathbf{f}(s), \boldsymbol{y}')| \leq \|\varphi\|_\infty \sum_{\boldsymbol{y}' \in \{0,1\}^n} \pi_{\boldsymbol{y}'}(s) = \|\varphi\|_\infty .$$

Taking the supremum over $s \in \mathcal{H}$ yields

$$\|\mathsf{P}\varphi\|_\infty \leq \|\varphi\|_\infty .$$

Thus, $\mathsf{P}\varphi$ is bounded and continuous whenever φ is bounded and continuous. By definition, P is therefore a Feller transition kernel (Meyn & Tweedie, 2009, Chapter 6).

Step 2: Average the state distributions and take a weak limit. Choose any initial state distribution ν; a deterministic initial state corresponds to a point mass. After k transitions, the state has distribution $\nu\mathsf{P}^k$. These round-specific distributions may fail to converge; for example, a periodic process can alternate between different regions of the state space. We therefore consider their arithmetic averages

$$\overline{\nu}_N := \frac{1}{N} \sum_{k=0}^{N-1} \nu\mathsf{P}^k.$$

Let $\mathcal{P}(\mathcal{H})$ denote the set of all probability distributions over states in $\mathcal{H}$. Since $\mathcal{H}$ is a compact metric space, every sequence in $\mathcal{P}(\mathcal{H})$ has a weakly convergent subsequence. In particular, $(\overline{\nu}_N)_{N\geq 1}$ has a subsequence satisfying

$$\overline{\nu}_{N_j} \Rightarrow \mu \qquad \text{for some } \mu \in \mathcal{P}(\mathcal{H}).$$

Here, weak convergence means that

$$\int_{\mathcal{H}} \varphi \, d\overline{\nu}_{N_j} \longrightarrow \int_{\mathcal{H}} \varphi \, d\mu \tag{31}$$

for every bounded continuous function φ. This characterization is the only property of weak convergence needed below.

Step 3: Show that the weak limit is invariant. First, applying one additional transition to the average shifts every term forward by one round. Linearity and $\nu\mathsf{P}^k\mathsf{P} = \nu\mathsf{P}^{k+1}$ yield

$$\overline{\nu}_N\mathsf{P} = \frac{1}{N} \sum_{k=0}^{N-1} \nu\mathsf{P}^{k+1} = \frac{1}{N} \sum_{k=1}^{N} \nu\mathsf{P}^k.$$

For any bounded continuous function φ, subtracting its expectation under $\overline{\nu}_N$ cancels all intermediate terms and leaves

$$\int_{\mathcal{H}} \varphi \, d(\overline{\nu}_N\mathsf{P}) - \int_{\mathcal{H}} \varphi \, d\overline{\nu}_N = \frac{1}{N} \left[\int_{\mathcal{H}} \varphi \, d(\nu\mathsf{P}^N) - \int_{\mathcal{H}} \varphi \, d\nu \right].$$

For every probability distribution η on $\mathcal{H}$, $|\int_{\mathcal{H}} \varphi \, d\eta| \leq \|\varphi\|_\infty$. The triangle inequality therefore yields

$$\left| \int_{\mathcal{H}} \varphi \, d(\overline{\nu}_N\mathsf{P}) - \int_{\mathcal{H}} \varphi \, d\overline{\nu}_N \right| \leq \frac{2\,\|\varphi\|_\infty}{N}. \tag{32}$$

Since $\|\varphi\|_\infty$ is finite and independent of N, the right-hand side converges to zero as $N \to \infty$. Thus, the effect of one additional transition on any bounded continuous expectation vanishes as $N \to \infty$.

For any state distribution η, the definition of the transition operator and the law of total expectation yield

$$\int_{\mathcal{H}} \varphi \, d(\eta\mathsf{P}) = \int_{\mathcal{H}} (\mathsf{P}\varphi) \, d\eta. \tag{33}$$

The left-hand side is the expected value of φ after one transition from η. The right-hand side averages the conditional next-state expectation $(\mathsf{P}\varphi)(s)$ over current states.

Second, Step 1 shows that $\mathsf{P}\varphi$ is bounded and continuous. Applying Equation (31) to $\mathsf{P}\varphi$ therefore yields

$$\int_{\mathcal{H}} (\mathsf{P}\varphi)\, d\overline{\nu}_{N_j} \longrightarrow \int_{\mathcal{H}} (\mathsf{P}\varphi)\, d\mu.$$

Using Equation (33) on both sides yields

$$\int_{\mathcal{H}} \varphi\, d(\overline{\nu}_{N_j}\mathsf{P}) \longrightarrow \int_{\mathcal{H}} \varphi\, d(\mu\mathsf{P}). \tag{34}$$

Finally, setting $N = N_j$ in Equation (32) yields

$$\left| \int_{\mathcal{H}} \varphi\, d(\overline{\nu}_{N_j}\mathsf{P}) - \int_{\mathcal{H}} \varphi\, d\overline{\nu}_{N_j} \right| \leq \frac{2\,\|\varphi\|_\infty}{N_j} \longrightarrow 0,$$

since $N_j \to \infty$. Equation (34) identifies the limit of the first expectation as $\int_{\mathcal{H}} \varphi\, d(\mu\mathsf{P})$, while Equation (31) identifies the limit of the second as $\int_{\mathcal{H}} \varphi\, d\mu$. Since their difference converges to zero, these limits are equal.

$$\int_{\mathcal{H}} \varphi\, d(\mu\mathsf{P}) = \int_{\mathcal{H}} \varphi\, d\mu$$

for every bounded continuous φ. Since bounded continuous functions distinguish probability distributions on a metric space (Meyn & Tweedie, 2009, Appendix D.5), $\mu\mathsf{P} = \mu$. This identity means that μ is an invariant distribution for the joint feature–outcome process. □

Interpretation. Proposition 2 guarantees that, despite repeated refitting, graph-coupled feature responses, and stochastic outcome updates, the process admits at least one invariant distribution. The proof constructs this distribution as a weak subsequential limit of the time-averaged state distributions. Thus, the process has a probabilistic regime that is preserved by deployment even when individual features and outcomes continue to change. Establishing uniqueness or convergence to this regime from an arbitrary initial distribution would require additional conditions on the feature dynamics.

C.3 Persistent outcome switching

Proposition 4 (Persistent outcome switching)**.** *Under the assumptions of Proposition 1 and $\varepsilon > 0$, every node changes its binary outcome infinitely often almost surely. This conclusion holds for any sequence of classifiers whose feature responses follow Equation* (19)*.*

Proof. We first show that every node has a uniformly positive conditional probability of switching at every round. This bound rules out a final switch for any node and then yields the graph-wide conclusion because the graph is finite. Fix an arbitrary node i.

Step 1: Bound the graph-dependent outcome logit. Let $B_x := \max_j |b_j| + c$. Proposition 1 yields $x_{j,t} \in [b_j - c, b_j + c]$ for every node j and round t. Hence

$$|x_{j,t}| \leq B_x, \qquad |\Delta x_{j,t}| = |x_{j,t+1} - x_{j,t}| \leq 2c.$$

If node i has degree $d_i > 0$, averaging over its neighbors preserves these bounds:

$$\left| \frac{1}{d_i} \sum_{j \in \mathcal{N}(i)} x_{j,t+1} \right| \leq B_x, \qquad \left| \frac{1}{d_i} \sum_{j \in \mathcal{N}(i)} \Delta x_{j,t} \right| \leq 2c.$$

The same inequalities hold for an isolated node because both neighbor means are defined as zero. Therefore, the logit in Equation (20) satisfies

$$\left| x_{i,t+1} + \mathrm{mean}_{j \in \mathcal{N}(i)}\, x_{j,t+1} + \mathrm{mean}_{j \in \mathcal{N}(i)}\, \Delta x_{j,t} \right| \leq 2B_x + 2c.$$

Step 2: Keep the resampled outcome probability away from zero and one. Set

$$\delta := \sigma(-(2B_x + 2c)) > 0.$$

Since the sigmoid is increasing and $\sigma(2B_x + 2c) = 1 - \delta$, the preceding logit bound implies

$$\delta \leq q_{i,t} \leq 1 - \delta$$

at every round, for every realized history and every classifier sequence.

Step 3: Obtain a uniform conditional probability of switching. Condition on the complete history through the feature update at round t. If $y_{i,t} = 0$, Equation (21) yields

$$\Pr(y_{i,t+1} \neq y_{i,t} \mid \text{history}) = \varepsilon q_{i,t} \geq \varepsilon\delta.$$

If $y_{i,t} = 1$, it follows that

$$\Pr(y_{i,t+1} \neq y_{i,t} \mid \text{history}) = \varepsilon(1 - q_{i,t}) \geq \varepsilon\delta.$$

Thus every round yields node i a switching probability of at least $\alpha := \varepsilon\delta > 0$, regardless of its current outcome or the preceding trajectory.

Step 4: Exclude a final switch. Fix a round N. Repeated conditioning shows that the probability that node i makes no switch during the next m rounds is at most

$$(1 - \alpha)^m.$$

This step holds without independence across rounds. It applies the conditional no-switch bound $1-\alpha$ after each possible history. Because $(1 - \alpha)^m \to 0$, the probability that node i never switches after round N is zero. If node i switched only finitely many times, then some such round N would follow its last switch. Taking the countable union over N therefore shows that node i switches infinitely often almost surely. Finally, the graph has finitely many nodes, so the union of the node-level failure events still has probability zero. Hence every node switches infinitely often almost surely. □

D Proofs for Synthetic Performance Results

Appendix C establishes the basic properties of the persistent classification process. We first prove the no-feature-response baseline result, then derive the fitting and state-sensitivity bounds used by the finite-horizon preservation theorem.

D.1 Equal expected loss without feature response

Proposition 5 (Equal expected loss without feature response)**.** *Fix the graph, features, and disjoint training and test sets. Initialize outcomes independently as $y_{i,0} \sim \text{Bernoulli}(q_i^0)$, and use independent node-round innovations with transition probability $(1-\varepsilon)y_{i,t} + \varepsilon q_i^0$. For a fixed representation and a deterministic fitting map using current training labels only, the refitted and fixed-classifier policies have equal expected test loss at every round and over every finite evaluation window.*

Proof. **Step 1: Preserve the outcome distribution.** For each node, a Bernoulli variable of mean q_i^0 remains Bernoulli with that mean after the specified outcome transition. Independent innovations preserve independence across nodes. Thus the product Bernoulli distribution with means q_i^0 is stationary.

Step 2: Separate fitting from test outcomes. The outcome processes on the training and test nodes are independent, conditional on the graph and features. Consequently, the fits based on the observed training labels $\mathbf{y}_{L,t}$ and $\mathbf{y}_{L,0}$ have the same marginal distribution and are each independent of the current test outcomes $\mathbf{y}_{U,t}$. Their joint distributions with the test outcomes coincide. Applying the same test-loss function to these joint distributions yields equal expectations at each round; averaging yields the window result. □

D.2 Explicit trajectory-separation bound and proof of Theorem 1

Condition on the graph, immutable features, reference vector, preprocessing, and common nonempty training and test sets. Suppose representation A satisfies, uniformly over the compact feature box,

$$\|\mathbf{z}_{A,i}(\mathbf{x})\|_2 \leq R_z, \qquad \|\mathbf{z}_{A,i}(\mathbf{x}) - \mathbf{z}_{A,i}(\widetilde{\mathbf{x}})\|_2 \leq L_z\|\mathbf{x} - \widetilde{\mathbf{x}}\|_\infty.$$

Let $m = |L|$, $n_U = |U|$, and $W = \sqrt{2\log 2/\lambda}$, and define

$$A_x = L_z(1 + R_z W/4), \quad H_x = WL_z + R_z A_x/\lambda, \quad H_y = R_z^2 n/(\lambda m).$$

With $\eta = ac/2$, set

$$M_x = 1 - a + \eta H_x, \qquad M_y = \eta H_y,$$

$$\mathbf{B} = \begin{bmatrix} M_x & M_y \\ \varepsilon(3M_x+1)/4 & 1-\varepsilon+3\varepsilon M_y/4 \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} \eta \\ 3\varepsilon\eta/4 \end{bmatrix}, \quad \mathbf{c}_* = \begin{bmatrix} H_x \\ H_y + R_z W n/n_U \end{bmatrix}.$$

For

$$\nu_t := \mathbb{E}\|\mathbf{h}_A(s_t^B) - \mathbf{h}_B(s_t^B)\|_\infty,$$

the trajectory-separation term in Theorem 1 is

$$C_T := \frac{1}{T}\sum_{t=1}^{T} \mathbf{c}_*^\top \sum_{k=0}^{t-1} \mathbf{B}^{t-1-k}\mathbf{g}\nu_k. \tag{35}$$

The coefficient 3 bounds the absolute row sums of $\mathbf{I} + 2\mathbf{N}$.

Step 1: Bound the change in fitted coefficients. All bounds condition on the graph, immutable features, preprocessing, and training/test memberships. Let $s = (\mathbf{x}, \mathbf{y})$ and $\widetilde{s} = (\widetilde{\mathbf{x}}, \widetilde{\mathbf{y}})$, and set $d_x = \|\mathbf{x} - \widetilde{\mathbf{x}}\|_\infty$ and $d_y = \|\mathbf{y} - \widetilde{\mathbf{y}}\|_1/n$. At any coefficient vector of norm at most W, subtracting the two training gradients yields the bound

$$\|\nabla\mathcal{L}_s(\mathbf{w}) - \nabla\mathcal{L}_{\widetilde{s}}(\mathbf{w})\|_2 \le A_x d_x + R_z n d_y/m.$$

Indeed, changing a row in $(\sigma(\mathbf{z}_i^\top\mathbf{w}) - y_i)\mathbf{z}_i$ contributes at most $L_z d_x + R_z W L_z d_x/4$, and changing its label contributes at most $R_z|y_i - \widetilde{y}_i|$. Strong monotonicity of the regularized gradient, applied at its two roots, therefore yields

$$\|\mathbf{w}(s) - \mathbf{w}(\widetilde{s})\|_2 \le \lambda^{-1}(A_x d_x + R_z n d_y/m).$$

Step 2: Transfer the bound to logits and loss. Insert and subtract $\mathbf{Z}(\mathbf{x})\mathbf{w}(\widetilde{s})$ to obtain

$$\|\mathbf{h}_A(s) - \mathbf{h}_A(\widetilde{s})\|_\infty \le H_x d_x + H_y d_y.$$

Logistic loss is 1-Lipschitz in its logit, and changing a binary target changes its loss by at most $|h_i| \le R_z W$. Thus

$$|\mathcal{R}_A(s) - \mathcal{R}_A(\widetilde{s})| \le H_x d_x + (H_y + R_z W n/n_U) d_y. \tag{36}$$

Step 3: Couple the evolving state trajectories. Write $D_t = \mathbb{E}\|\mathbf{x}_t^A - \mathbf{x}_t^B\|_\infty$ and $E_t = \mathbb{E}\|\mathbf{y}_t^A - \mathbf{y}_t^B\|_1/n$. Inserting A's refitted logits on B's state into the response difference yields

$$D_{t+1} \le M_x D_t + M_y E_t + \eta\nu_t.$$

Under shared uniform draws, two Bernoulli variables disagree with conditional probability equal to the absolute difference of their parameters. The outcome-resampling logit is $(\mathbf{I} + 2\mathbf{N})\mathbf{x}_{t+1} - \mathbf{N}\mathbf{x}_t$, with $\|\mathbf{N}\|_\infty \le 1$. Hence

$$E_{t+1} \le (1-\varepsilon)E_t + \frac{\varepsilon}{4}(3D_{t+1} + D_t).$$

Step 4: Accumulate discrepancies and compare risks. Substitution yields the nonnegative matrix recurrence $[D_{t+1}, E_{t+1}]^\top \le \mathbf{B}[D_t, E_t]^\top + \mathbf{g}\nu_t$. Iterating from the common initial state yields the inner sum in Equation (12). Add and subtract $\mathcal{R}_A(s_t^B)$ in the performance difference and apply Equation (36). In particular,

$$J_T(A) - J_T(B) + \Gamma_T = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\mathcal{R}_A(s_t^A) - \mathcal{R}_A(s_t^B)].$$

Bounding the absolute value of this expression proves the theorem. If the initial graph is random, the conditional bound is averaged over that graph as well. Its random constants remain inside the expectation.

D.3 A GRAPH-RESOLVED TRAJECTORY BOUND

For the propagated representation, a componentwise bound retains the graph paths suppressed by the scalar constants in Theorem 1. Absorb the fixed column scales into the coefficients, set $\mathbf{P} = \mathbf{S}^K$, and write the fixed-classifier logits as $\mathbf{h}_F(\mathbf{x}) = w_r\mathbf{P}\mathbf{r} + w_x\mathbf{P}\mathbf{x} + w_0\mathbf{1}$. Define the componentwise discrepancies

$$\mathbf{d}_t = |\mathbf{x}_t^{\text{refit}} - \mathbf{x}_t^{\text{fixed}}|, \quad \mathbf{e}_t = |\mathbf{y}_t^{\text{refit}} - \mathbf{y}_t^{\text{fixed}}|, \quad \boldsymbol{\delta}_t = |\mathbf{h}_t^{\text{refit}}(\mathbf{x}_t^{\text{refit}}) - \mathbf{h}_F(\mathbf{x}_t^{\text{refit}})|.$$

Set $\mathbf{Q}_N = \mathbf{I}+2\mathbf{N}$, $\mathbf{M} = (1-a)\mathbf{I}+(ac/2)|w_x|\mathbf{P}$, and $\eta = ac/2$. Under shared outcome randomness, nonnegativity of $\mathbf{P}$ and the $1/4$-Lipschitz property of the sigmoid yield

$$\mathbf{d}_{t+1} \le \mathbf{M}\mathbf{d}_t + \eta\boldsymbol{\delta}_t, \qquad \mathbb{E}\mathbf{e}_{t+1} \le (1-\varepsilon)\mathbb{E}\mathbf{e}_t + \frac{\varepsilon}{4}(\mathbf{Q}_N\mathbb{E}\mathbf{d}_{t+1} + \mathbf{N}\mathbb{E}\mathbf{d}_t).$$

With $\mathbf{u}_t = [\mathbb{E}\mathbf{d}_t^\top, \mathbb{E}\mathbf{e}_t^\top]^\top$, define

$$\mathbf{B}_F = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \varepsilon(\mathbf{Q}_N\mathbf{M} + \mathbf{N})/4 & (1-\varepsilon)\mathbf{I} \end{bmatrix}, \qquad \mathbf{G}_F = \begin{bmatrix} \eta\mathbf{I} \\ \varepsilon\eta\mathbf{Q}_N/4 \end{bmatrix}.$$

Since $\mathbf{u}_0 = \mathbf{0}$, iteration gives $\mathbf{u}_t \le \sum_{k=0}^{t-1} \mathbf{B}_F^{t-1-k}\mathbf{G}_F\mathbb{E}\boldsymbol{\delta}_k$. Let $\mathbf{r}_U$ assign weight $1/|U|$ to each test node and zero elsewhere, and define

$$\mathbf{k}_F = |w_r|\,|\mathbf{P}\mathbf{r}| + |w_x|\mathbf{P}(|\mathbf{b}| + c\mathbf{1}) + |w_0|\mathbf{1}, \qquad \mathbf{c}_F^\top = [\,\mathbf{r}_U^\top|w_x|\mathbf{P} \mid \mathbf{r}_U^\top \operatorname{diag}(\mathbf{k}_F)\,].$$

The 1-Lipschitz property of logistic loss and $|\mathbf{h}_F| \le \mathbf{k}_F$ then bound the absolute expected induced-state contribution by

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{k=0}^{t-1} \mathbf{c}_F^\top\mathbf{B}_F^{t-1-k}\mathbf{G}_F\mathbb{E}\boldsymbol{\delta}_k. \tag{37}$$

This bound retains graph paths and their alignment with discrepancies; its absolute values leave the direction of the induced-state contribution undetermined.

E DATA GENERATION AND EXPERIMENTAL DETAILS

Having completed the supporting theory, we now specify the synthetic generator, coupling, evaluation, and comparison settings used in the empirical study. These settings are also the basis of the neural robustness experiments in Appendix F.

E.1 INITIAL GRAPH AND PREPROCESSING

We generate $n = 400$ latent group indicators independently as $g_i \sim \text{Bernoulli}(1/2)$; redraw the group vector if a group contains fewer than 50 nodes. Conditional on these indicators, connect distinct unordered node pairs independently with probability 0.04 for equal group indicators and 0.01 otherwise. Generate raw attributes

$$r_i^{\text{raw}} = 0.5(2g_i - 1) + \xi_i, \qquad x_{i,0}^{\text{raw}} = 0.5(2g_i - 1) + \zeta_i,$$

where ξ_i, ζ_i are independent standard normal variables drawn once. The first coordinate is immutable. Latent groups determine the feature-generating means.

Stratified 60/20/20 splits are based on latent groups, with memberships fixed across rounds. For each feature coordinate k, fit

$$\mu_k = \frac{1}{|L|}\sum_{i\in L} X_{i,k,0}^{\text{raw}}, \qquad s_k = \left[\frac{1}{|L|}\sum_{i\in L}(X_{i,k,0}^{\text{raw}} - \mu_k)^2\right]^{1/2}.$$

We replace a zero scale by one. Standardize all nodes using $X_{i,k,0} = (X_{i,k,0}^{\text{raw}} - \mu_k)/s_k$. We freeze these statistics and set $b_i = x_{i,0}$. Subsequent feature evolution uses the standardized coordinates directly. Neither attributes nor graph structure are regenerated to maintain correlations with changing outcomes. Primary parameters are declared before test evaluation. The validation set is reserved for declared development use.

After standardization, we generate initial outcomes independently as

$$q_i^0 = \sigma(x_{i,0} + (\boldsymbol{N}\boldsymbol{x}_0)_i)\,, \qquad y_{i,0} \sim \text{Bernoulli}(q_i^0).$$

The operator $\boldsymbol{N}$ averages original neighbors without self-loops. Each realization has independent initial outcome draws; memberships remain fixed, and every outcome draw is retained regardless of class balance. Only the outcomes on training nodes are observed as labels and enter the initial fit. These draws are independent of subsequent outcome uniforms. The original initialization, $y_{i,0} = g_i$, is retained as a sensitivity study. The primary initialization is stationary for the independent-outcome process with fixed features. Feature response may make the initial distribution nonstationary.

E.2 Representation and transition details

The primary R/G/C comparison uses $K = 2$. In Equation (14), the diagonal matrix $\mathbf{D}_K$ matches each propagated column's initial training root mean square to that of the corresponding original column and remains fixed across rounds. A zero propagated column receives multiplier one. The unscaled comparison uses $\mathbf{D}_K = \boldsymbol{I}$. The intercept is appended after propagation; the graph operator leaves it unchanged. Consequently, R and G have three readout coordinates and C has five. Every coordinate, including the intercept, receives the quadratic penalty in Equation (16).

For completeness, the feature and outcome updates induce the deployed transition kernel

$$\mathsf{K}_{\boldsymbol{w}}(C \mid s) = \sum_{\boldsymbol{y}' \in \{0,1\}^n} \mathbf{1}\{((V, E, \mathcal{F}_X(\boldsymbol{X}, \boldsymbol{w})), \boldsymbol{y}') \in C\} \prod_{i=1}^{n} \text{Bern}(\{y_i'\}; \rho_i(s, \boldsymbol{w}))\,. \tag{38}$$

Here $\text{Bern}(\{y_i'\}; \rho)$ is the Bernoulli probability mass at y_i', and $\rho_i(s, \boldsymbol{w})$ is obtained from Equation (21) after applying the feature update $\mathcal{F}_X(\boldsymbol{X}, \boldsymbol{w})$. Conditional on $(s, \boldsymbol{w})$, the feature update is deterministic and the outcome draws are independent.

E.3 Experimental conditions, coupling, and evaluation

The independent replication uses twenty independently generated datasets and five independent outcome-process realizations per dataset. The three representations, two policies, and adjustment values $a \in \{0, 0.1\}$ produce 1,200 trajectories. Each runs for 1,000 rounds with $K = 2$ for G/C, $\lambda = 0.1$, $c = 1$, positive response, and $\varepsilon = 0.1$. The two additional outcome mechanisms use

$$q_t = \sigma\big(b_o x_{t+1} + b_n N x_{t+1} + N(x_{t+1} - x_t)\big), \qquad (b_o, b_n) \in \{(2, 0), (0, 2)\},$$

with initial means $\sigma(b_o x_0 + b_n N x_0)$. They add 2,400 trajectories to the mixed-mechanism study. The neighboring-change coefficient remains one.

Within each dataset/realization, experimental conditions share the uniforms used for the initial outcome draws and sequences of independent node-round outcome uniforms. Each condition evaluates its own transition probability and sets $y_{i,t+1} = \mathbf{1}\{U_{i,t} < \rho_{i,t}\}$. Sharing randomness pairs comparisons without forcing their outcomes to coincide. Initial uniforms are independent of subsequent outcome draws.

The primary statistic averages after-refitting test log loss over rounds 601–1000. Reported measurements also include pre-refit loss, AUROC, class prevalence, feature movement, coefficient movement, and outcome switching. Single-class snapshots have undefined AUROC. Temporal analyses compare windows 201–400, 401–600, 601–800, and 801–1000. These descriptive windows carry no mixing-time interpretation. The main comparisons use the prespecified late window, avoiding selection based on favorable performance.

E.4 Dataset-level uncertainty

Let $M_{d,r}$ be a trajectory-level summary for dataset d and outcome realization r, at a declared stage and window. Compute

$$\overline{M}_d = \frac{1}{5} \sum_{r=1}^{5} M_{d,r}, \qquad \widehat{M} = \frac{1}{20} \sum_{d=1}^{20} \overline{M}_d.$$

The five runs share initial graph structure and features. Averaging within dataset separates repeated outcome randomness from the 20 independent initial environments. Neither the 100 trajectories nor their individual rounds are treated as 100 independent datasets.

For example, define the paired log-loss difference

$$D_d = \overline{M}_{\text{refit},d} - \overline{M}_{\text{fixed},d}, \qquad \widehat{D} = \frac{1}{20}\sum_{d=1}^{20} D_d.$$

Negative values favor refitting. We resample 20 dataset indices with replacement, preserving each pair, and average the selected differences. Repeat 10,000 times. The 2.5th and 97.5th percentiles yield an approximate 95% percentile-bootstrap interval. The same dataset-level procedure applies to individual-condition means. Dataset-level values show heterogeneous effects. These intervals characterize average performance under the synthetic generator and finite replication budget. Graph-specific and application-level guarantees lie outside their scope.

E.5 State-trajectory Analysis

Figure 3 reports the complete state-trajectory analysis. Both policies change the mutable features from their common initial values. Refitting produces a distinct feature trajectory, and the paired policies continue to differ in both features and outcomes. Outcome switching also continues under both policies after aggregate feature displacement has levelled off. These curves show empirical stabilization of selected summaries. Convergence of an individual state or to an invariant distribution remains unverified.

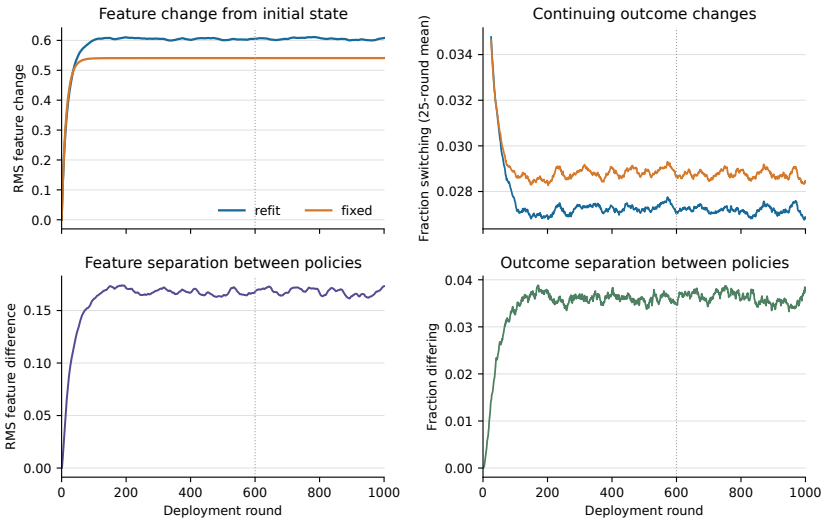

Figure 3: Expanded state analysis for C under positive feature response. The curves summarize twenty independently generated datasets, each with five coupled outcome-update realizations. We first average the five realizations within each dataset and then average equally across datasets. The top panels show RMS mutable-feature change from the initial state and the fraction of nodes switching outcomes between consecutive rounds, with the latter displayed as a trailing 25-round mean. The bottom panels show RMS feature difference and the fraction of differing outcomes between the paired refitting and fixed-classifier states. Within each pair, the two policies start from the same state and use the same outcome-update random numbers. The dotted line marks the start of the rounds 601–1000 evaluation window.

E.6 Initialization and Scale Comparisons

Two comparisons assess sensitivity to initialization and feature scaling. First, replacing the initialization $y_{i,0} = g_i$ by draws from the baseline outcome mechanism changes the unscaled $K = 2$

refitted-minus-fixed late loss from $+0.060152$ to -0.051895. The latter exploratory interval is $[-0.069683, -0.035284]$. The fixed-classifier comparator changes substantially. Refitting trajectories become nearly identical late in these coupled runs. Individual outcomes may continue evolving after the coupled trajectories coincide. The primary replication therefore uses baseline-consistent initialization.

Second, a feature rescaling changes the effective ridge penalty unless that penalty is transformed too. If $\widetilde{\mathbf{Z}} = \mathbf{Z}\mathbf{D}$ and $\mathbf{w} = \mathbf{D}\mathbf{v}$, then $\widetilde{\mathbf{Z}}\mathbf{v} = \mathbf{Z}\mathbf{w}$, and preserving $\lambda\|\mathbf{w}\|^2/2$ requires penalty $\lambda\mathbf{v}^\top\mathbf{D}^2\mathbf{v}/2$. Equivalent parameterizations yielded maximum logit difference 2.46×10^{-10} and no outcome mismatches in twenty matched 1,000-round pairs. RMS scaling with an unchanged isotropic penalty is therefore a substantive fitting intervention. The R/G/C comparison declares its scales and penalty explicitly.

On common environment states, linear reconstruction error for the outcome refresh logit increases from approximately 0.090 to 0.198 after propagation, despite RMS compensation. This analysis concerns a restricted linear readout. A general loss of information under every graph representation remains unestablished. The combined representation therefore retains original features alongside propagated features.

F End-to-End GCN Robustness Experiments

Appendix E specifies the shared synthetic environment and evaluation design. The primary experiments use fixed representation maps so that every logistic refit is unique and its numerical error can be certified. This section changes the predictor while retaining the synthetic environment. Its purpose is to test whether the empirical comparison between a fixed classifier and repeated refitting survives end-to-end nonlinear training. The nonconvex neural objectives lack the uniqueness and strong-convexity properties assumed by the synthetic results in Sections 3 and 3. Accordingly, the results below provide an empirical architecture robustness analysis. The guarantees remain specific to the convex setting.

F.1 Experimental settings

Predictors. Let $\phi(z) = \max(z, 0)$ act coordinatewise, and let every hidden layer have width 16. The two-layer multilayer perceptron (MLP) uses

$$\boldsymbol{H}_{\mathrm{R}}^{(1)} = \phi(\boldsymbol{X}\boldsymbol{W}_{\mathrm{R}}^{(1)} + \mathbf{1}\boldsymbol{b}_{\mathrm{R}}^{(1)\top}), \qquad \boldsymbol{H}_{\mathrm{R}}^{(2)} = \phi(\boldsymbol{H}_{\mathrm{R}}^{(1)}\boldsymbol{W}_{\mathrm{R}}^{(2)} + \mathbf{1}\boldsymbol{b}_{\mathrm{R}}^{(2)\top}).$$

The two-layer GCN uses

$$\boldsymbol{H}_{\mathrm{G}}^{(1)} = \phi(\boldsymbol{S}\boldsymbol{X}\boldsymbol{W}_{\mathrm{G}}^{(1)} + \mathbf{1}\boldsymbol{b}_{\mathrm{G}}^{(1)\top}), \qquad \boldsymbol{H}_{\mathrm{G}}^{(2)} = \phi(\boldsymbol{S}\boldsymbol{H}_{\mathrm{G}}^{(1)}\boldsymbol{W}_{\mathrm{G}}^{(2)} + \mathbf{1}\boldsymbol{b}_{\mathrm{G}}^{(2)\top}).$$

Their logit vectors are

$$\boldsymbol{h}_{\mathrm{R}} = \boldsymbol{H}_{\mathrm{R}}^{(2)}\boldsymbol{w}_{\mathrm{R}} + b_{\mathrm{R}}\mathbf{1}, \qquad \boldsymbol{h}_{\mathrm{G}} = \boldsymbol{H}_{\mathrm{G}}^{(2)}\boldsymbol{w}_{\mathrm{G}} + b_{\mathrm{G}}\mathbf{1}.$$

The skip-GCN retains a direct raw-feature channel,

$$\boldsymbol{h}_{\mathrm{SG}} = \boldsymbol{H}_{\mathrm{G}}^{(2)}\boldsymbol{w}_{\mathrm{G}} + \boldsymbol{X}\boldsymbol{v} + b_{\mathrm{SG}}\mathbf{1}.$$

Thus the MLP and GCN each have 337 fitted scalar parameters; the skip-GCN has 339. Predicted probabilities are obtained by applying the logistic function to the corresponding logits. The one-layer sensitivity removes the second hidden layer from each architecture. The dropout rate is zero.

Fitting and deployment. We minimize training-node binary cross-entropy plus $\lambda\|\boldsymbol{\theta}\|_2^2/2$ over every neural parameter, including biases. The primary neural setting uses $\lambda = 0.01$, deterministic full-batch Adam with learning rate 0.03, 400 initial updates, and 12 warm-started updates after each environmental transition. The fixed-classifier policy keeps the initial parameters. The refitting policy applies the declared updates to current training features and labels at every round. These finite neural updates define the policy.

The cold-start sensitivity reinitializes the predictor from the same fixed random parameters and applies 80 updates at every round, including the initial fit. The weight-decay comparison uses $\lambda \in \{0.005, 0.01, 0.02\}$. Fitting uses training labels exclusively.

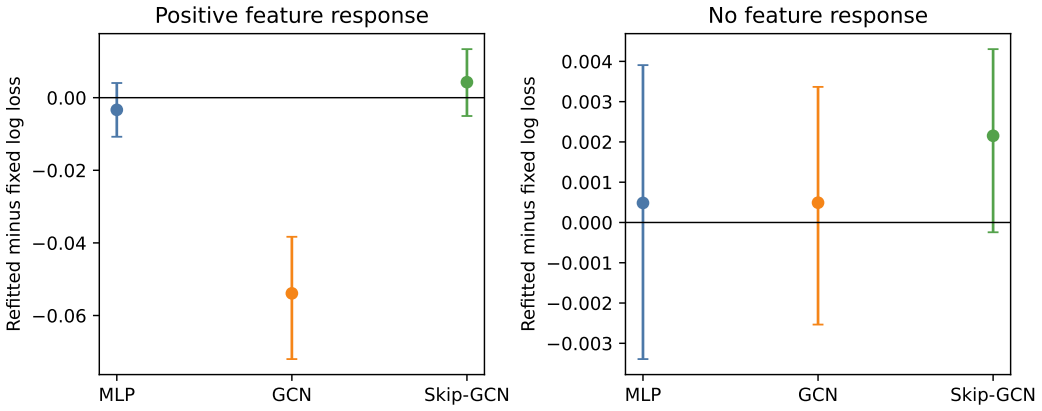

Figure 4: Neural refitted-minus-fixed log-loss gaps with paired 95% dataset-bootstrap intervals. Negative values favor refitting. The GCN benefit appears with positive feature response and disappears when the prediction-to-feature pathway is removed.

Environment and evaluation. The study uses the same twenty independently generated graphs as the fixed-filter study and the first outcome realization for each graph. All cells therefore contain 20 independent graphs. Within each graph, architectures and policies share the initial feature matrix, the mechanism-specific initial-outcome uniforms, and subsequent outcome uniforms. Each graph has separately fitted parameters. Every trajectory has 1,000 rounds. The primary summary averages rounds 601–1000. Intervals use the paired dataset bootstrap in Appendix E.4. The positive-response setting has $a = 0.1$. The matched $a = 0$ setting removes the prediction-to-feature pathway. We also vary the outcome mechanism among the own-feature, mixed, and neighboring- feature mechanisms from Section 4.

The primary quantity is the refitted-minus-fixed test log-loss difference. We additionally report AUROC, calibration error, Brier score, conditional outcome entropy, feature displacement, outcome switching, and prediction confidence. On the refitting-induced state, we evaluate the initial neural predictor to decompose the policy gap as in Equation (9). This comparison holds both the architecture and the realized state fixed when measuring the fitting contribution.

F.2 Experimental results

Primary comparison and no-feature-response baseline. Table 2 shows that refitting the GCN reduces late test log loss by 0.05390, with paired 95% interval $[-0.07200, -0.03832]$. The MLP and skip-GCN gaps are respectively -0.00335 and $+0.00427$; both intervals contain zero. Under $a = 0$, the corresponding gaps are $+0.00049$, $+0.00049$, and $+0.00215$, and all three intervals contain zero. Every measured induced-state contribution is then exactly zero. The positive-response minus no-feature difference in GCN policy gaps is -0.05439 with interval $[-0.07129, -0.04015]$.

The update policy also reverses the ordering of the two graph models. Under the fixed policy, skip-GCN minus GCN loss is -0.03453 with interval $[-0.05285, -0.01579]$. Under refitting it is $+0.02364$ with interval $[0.01194, 0.03503]$. The paired policy interaction is therefore $+0.05817$ with interval $[0.04131, 0.07664]$: retaining a raw-feature skip improves the fixed classifier; the repeatedly refitted plain GCN has lower trajectory loss.

Where the GCN improvement arises. The common-state fitting contribution to the GCN gap is -0.01586 with interval $[-0.02542, -0.00794]$. The larger induced-state contribution is -0.03803 with interval $[-0.05119, -0.02772]$. Figure 5 contrasts this reinforcement with the small, inconclusive components for the other architectures. Refitting the GCN raises mean AUROC from 0.8089 to 0.8564, reduces conditional outcome entropy from 0.1244 to 0.1144, and reduces the outcome-switch rate from 2.97% to 2.69%. Its mutable-feature standard deviation increases from 1.2485 to 1.4142. Together, these measurements locate the negative induced-state term: the refitted GCN

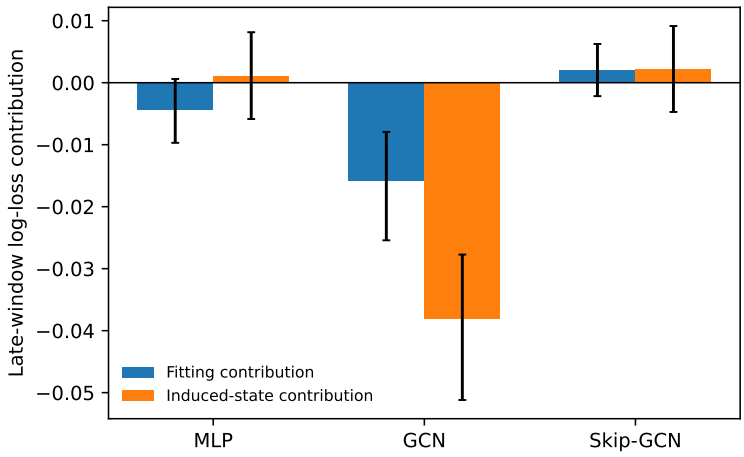

Figure 5: Common-state decomposition of the positive-response policy gap. Bars show late-window means and paired 95% intervals. The GCN's fitting and induced-state contributions have the same favorable sign.

Table 5: Sensitivity of late refitted-minus-fixed log loss. Brackets report paired 95% dataset-bootstrap intervals. Unless a row states otherwise, $a = 0.1$, $\lambda = 0.01$, two hidden layers, and warm-started refitting are used.

Setting	MLP	GCN	Skip-GCN
No feature response	$+.0005$ $[-.0034, .0039]$	$+.0005$ $[-.0025, .0034]$	$+.0022$ $[-.0002, .0043]$
Own-feature outcome	$-.0158$ $[-.0265, -.0060]$	$-.1197$ $[-.1717, -.0712]$	$-.0021$ $[-.0124, .0064]$
Neighbor-feature outcome	$-.0085$ $[-.0165, -.0014]$	$-.0475$ $[-.0600, -.0356]$	$-.0208$ $[-.0316, -.0107]$
$\lambda = 0.005$	$-.0022$ $[-.0111, .0064]$	$-.0818$ $[-.1168, -.0518]$	$-.0315$ $[-.0431, -.0204]$
$\lambda = 0.02$	$-.0135$ $[-.0223, -.0047]$	$-.0757$ $[-.0978, -.0546]$	$-.0021$ $[-.0090, .0045]$
Cold-start fitting	$-.0040$ $[-.0118, .0039]$	$-.0563$ $[-.0741, -.0408]$	$-.0353$ $[-.0430, -.0276]$
Cold start, no feature response	$+.0008$ $[-.0030, .0043]$	$+.0004$ $[-.0026, .0032]$	$+.0009$ $[-.0011, .0027]$
One hidden layer	$-.0030$ $[-.0103, .0043]$	$-.0713$ $[-.0827, -.0608]$	$-.0227$ $[-.0311, -.0155]$

maintains greater feature separation and induces less uncertain outcomes under the stated response mechanism.

Mechanism, regularization, initialization, and depth. Table 5 establishes that the plain GCN refitting benefit persists across both endpoint outcome mechanisms, all three weight-decay values, cold starts, and one or two hidden layers. Its warm- minus-cold policy-gap difference is $+0.00239$ with interval $[-0.00027, 0.00495]$. Warm-started optimization alone therefore cannot explain it.

Architecture levels remain mechanism dependent. Under the own-feature outcome mechanism, refitted losses are 0.2155, 0.4971, and 0.2301 for MLP, GCN, and skip-GCN. Under the neighbor-feature mechanism they are 0.6747, 0.5734, and 0.6180. The model with access to the relevant channel is favored. Adding a channel may increase deployment loss.

The skip-GCN is more sensitive to fitting and depth. Its warm two-layer gap is inconclusive, while its cold-start gap is -0.03527 and its one-layer gap is -0.02269. The warm-minus-cold interaction is $+0.03954$ with interval $[0.03253, 0.04661]$, and the two-minus-one-layer interaction is $+0.02696$ with interval $[0.01943, 0.03431]$. Its benefit also appears at $\lambda = 0.005$ and disappears at 0.01 and 0.02. Neural optimization and propagation depth are therefore distinct components of the deployed update policy.

F.3 Implications

These experiments extend the paper's empirical conclusion beyond fixed graph filters. The central result concerns how architecture and refitting jointly determine the environment encountered later. In the mixed mechanism, the fixed policy favors skip-GCN over GCN, while repeated refitting favors

GCN over skip-GCN. Common-state evaluation attributes the GCN advantage to both improved fitting and a larger, favorable induced-state contribution.

The comparisons identify when this conclusion changes. Removing feature response eliminates the policy gaps. Changing which information governs outcomes changes the architecture ordering. Weight decay, initialization, and propagation depth materially change the skip-GCN result, while the plain GCN refitting benefit retains its sign. These findings support evaluating graph architecture, fitting rule, and environmental response as one deployed system.

The exact-refitting theory remains confined to the convex objectives analyzed in the main results. The neural comparison supplies an empirical counterpart: the loss decomposition is an identity for predictors produced by a potentially nonunique fitting map and finite optimization. The study uses one outcome realization on each of twenty independent graphs. Its intervals measure variation across those graphs. The higher-budget fixed-filter study remains the primary replication across outcome randomness.

G Application Construction and Evaluation

The preceding empirical appendices concern the designed synthetic environment. This section specifies the distinct user–product application, including its information restrictions, action space, and response assumptions.

G.1 Observations, text, and information access

For each user–product pair, preprocessing retains the latest recorded review and partitions deduplicated pairs by a deterministic ordering into 70% training, 15% validation, and 15% test. This fixed offline split measures held-out generalization within the sampled period. The training set has 2,082,885 pairs and the test set has 446,334. An edge records a training interaction regardless of its rating; the binary label is rating at least four. Validation supports model development, while test ratings enter reported evaluation. User profiles, audience selection, candidate generation, and fitting use training-review text exclusively.

The frozen sentence encoder is all-MiniLM-L6-v2.[1] Product title and retained description text are split into at most 240 wordpiece chunks, encoded separately, averaged with token-count weights, and normalized. A Gaussian 384×64 matrix is orthonormalized and multiplied by $\sqrt{384/64}$, producing a fixed random projection. The same projection is used for review and product embeddings. The declared random projection remains fixed throughout. Actual edited text is re-encoded, and the experiment excludes unconstrained embedding perturbations.

For user u, $\mathbf{r}_u^+$ and $\mathbf{r}_u^-$ are separate means of positive and negative training-review features. Missing groups receive a zero vector and an availability indicator. A training pair excludes its own review from these means and indicators. For validation, test, and audience queries, only the training history is used. The fixed summaries encode past reported experiences and remain unchanged across rounds.

G.2 Graph operators and the pair classifier

Let $\mathbf{B} \in \{0,1\}^{n_u \times n_p}$ be training incidence for n_u users and n_p products, with degrees d_u, d_i. The feature matrix $\mathbf{X} \in \mathbb{R}^{n_p \times 64}$ has row $\mathbf{x}_i^\top$. The matrices $\mathbf{T}_B, \mathbf{M}$ are $n_u \times n_p$, and $\mathbf{H}$ is $n_p \times n_p$. Define

$$\mathbf{T}_B = \operatorname{diag}((d_u+1)^{-1/2})\mathbf{B}\operatorname{diag}((d_i+1)^{-1/2}), \quad \mathbf{H} = \mathbf{T}_B^\top \mathbf{T}_B + \operatorname{diag}((d_i+1)^{-2}),$$

$$\mathbf{M} = \operatorname{diag}(\max(d_u,1)^{-1})\mathbf{B}, \qquad \mathbf{g}_i = [(\mathbf{HX})_{i,:}]^\top, \quad \mathbf{m}_u = [(\mathbf{MX})_{u,:}]^\top.$$

The matrix $\mathbf{H}$ is the product block of two normalized bipartite propagation steps with one self-loop per node and zero initial user features in this propagation channel. Review profiles are supplied separately. $\mathbf{M}$ averages current product features over each user's fixed training neighbors and returns zero for users without such neighbors.

[1] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

With coordinatewise multiplication denoted by $\odot$, the three pair designs, before fixed RMS scaling, are column vectors. Let col stack its arguments vertically, including scalar indicators. Then

$$\boldsymbol{\psi}^{\mathrm{R}}_{ui} = \mathrm{col}(\mathbf{r}^+_u, \mathbf{r}^-_u, \mathbf{x}_i, \mathbf{r}^+_u \odot \mathbf{x}_i, \mathbf{r}^-_u \odot \mathbf{x}_i, b^+_u, b^-_u, \mathbf{m}_u, 1),$$
$$\boldsymbol{\psi}^{\mathrm{G}}_{ui} = \mathrm{col}(\mathbf{r}^+_u, \mathbf{r}^-_u, \mathbf{g}_i, \mathbf{r}^+_u \odot \mathbf{g}_i, \mathbf{r}^-_u \odot \mathbf{g}_i, b^+_u, b^-_u, \mathbf{m}_u, 1),$$
$$\boldsymbol{\psi}^{\mathrm{C}}_{ui} = \mathrm{col}(\mathbf{r}^+_u, \mathbf{r}^-_u, \mathbf{x}_i, \mathbf{g}_i, \mathbf{r}^+_u \odot \mathbf{x}_i, \mathbf{r}^-_u \odot \mathbf{x}_i, \mathbf{r}^+_u \odot \mathbf{g}_i, \mathbf{r}^-_u \odot \mathbf{g}_i, b^+_u, b^-_u, \mathbf{m}_u, 1).$$

The indicators b^+_u, b^-_u record nonempty review groups. The dimensions are 387, 387, and 579. Each non-intercept coordinate is divided by its initial training RMS, floored at 10^{-6}; these scales are fixed. All coefficients, including the intercept, receive penalty $\lambda = 0.001$. The probability of a favorable pair rating is $\sigma(\boldsymbol{\psi}^\top_{ui}\mathbf{v})$, where $\mathbf{v}$ includes the fixed scaling. R uses raw product features and graph-derived user histories and omits product propagation. The extra C coefficients also make parameter count vary across representations.

G.3 Audiences, candidate descriptions, and adoption

Each product has 100 distinct training-supported audience users sampled once, excluding its own training reviewers. Audiences remain identical across model comparisons. The eligibility threshold $\tau = 0.8483823127627923$ was fixed from the median initial R score on nine calibration products and retained in subsequent comparisons. The small calibration set and the reuse of this threshold across models make it a declared platform rule specific to this simulation. Its use in industry requires separate evidence. Scores average continuous probabilities before the eligibility threshold is applied. The repeated study uses zero additional threshold margin.

The broader screen selected 243 products with training connectivity, editable text, and initial scores within 0.03 below threshold for all three static-profile models used during screening. Source review cleared 124 products and held 119. Dynamic-history models reassess initial eligibility on this same cohort. It is a selected recourse cohort, so the reported results describe this cohort. Its 1,820 warm test pairs are fixed across conditions and include products that never edit; calling it an adopter cohort would change its meaning. Full warm evaluation uses 144,991 pairs. Secondary scopes retain all 5,236 test pairs for cleared products and all 446,334 test pairs, including cold users/products.

Each model searches ten common and ten model-guided candidate slots per product, plus the original description. Candidate sets generated with the static-profile models remain fixed in the dynamic study. Source facts, qualifications, and immutable text must be preserved. Candidates are screened for empty text, duplicates, and the word ceiling $\max(80, \lceil 1.5 \,\text{original word count} \rceil)$. Factual review uses separate generation and review prompts, and every selected rewrite receives an additional source-based review. The procedure uses automated and assistant review without independent human factual certification. The interpretation of unchanged historical rating labels is conditional on preserving the product represented by its description.

At each round, an ineligible product searches candidates using the current classifier and environment state. Cost is token-level Levenshtein distance from its current text, divided by its current token count (or one for empty text). The least-cost reviewed candidate with audience score at least τ is chosen, with lexicographic tie-breaking. Edit cost is unrestricted. Eligible products remain unchanged, and ineligible products with no threshold-crossing candidate also remain unchanged. All recourse recommendations are adopted, and the corresponding description edits are implemented fully and simultaneously. Recompute $\mathbf{HX}$ and $\mathbf{MX}$, then either retain the initial coefficients or refit once. Graph edges, historical ratings, and review summaries remain fixed.

Excluding own reviewers from each audience implies $M_{ui} = 0$ for its members. A unilateral edit therefore leaves those users' history means unchanged; other products' simultaneous edits can change them. The application includes this indirect route explicitly when evaluating post-adoption predictions. New purchasing and review behavior lie outside the simulation.

G.4 Assumptions and scope of empirical evidence

In the synthetic study, the restoring feature response keeps the feature state compact, while the resulting bounded outcome-update logits keep the resampling probabilities away from zero and one. These properties characterize the stipulated mechanism. Physical and behavioral relationships re-

quire separate evidence. Independent conditional outcome innovations separate individual randomness from graph-mediated dependence. The outcome-mechanism comparisons assess sensitivity to which feature channels matter within this family.

In the application, full adoption and fixed historical rating labels define a structured study of presentation changes. A fixed review label records a past evaluation. Future customer response remains unobserved. Fixed candidate sets make reproducible policy comparisons possible and account for early stopping. They exclude further improvements from newly generated recourse recommendations. Threshold calibration, category selection, and model development used this application repeatedly. The results are therefore descriptive for the completed cohort, without a claim of a fresh confirmatory application test. The independently generated synthetic replication supplies the study's dataset-level uncertainty estimates.

H Proofs for the Description-Recourse Results

The result on recourse recommendations uses the fixed graph, historical rating labels, audiences, and candidate descriptions specified in Section 2.2 and Appendix G. The loss-increment bound is an algebraic statement for fixed labels.

H.1 Proof of Proposition 3

Proof. **Step 1: Bound each score change.** Since the sigmoid is $1/4$-Lipschitz, averaging the absolute pairwise probability changes yields $|g_i^{\mathrm{refit}}(k) - g_i^{\mathrm{fixed}}(k)| \le b_i(k)$.

Step 2: Preserve the selection. The strict margin condition puts both scores on the same side of τ for the current description and every candidate. The current eligibility and the set of threshold-crossing candidates therefore agree. With unchanged costs and tie-breaking, the fixed and refitted policies also select the same recourse recommendation. □

H.2 Empirical loss increment

Lemma 2 (Empirical loss increment)**.** *Let $\mathbf{z}, \mathbf{d}, \mathbf{y} \in \mathbb{R}^m$ collect base logits, logit changes, and fixed binary labels. Define*

$$A(\mathbf{z}, \mathbf{d}) \coloneqq \frac{1}{m} \sum_{j=1}^{m} (\sigma(z_j) - y_j) d_j, \qquad Q(\mathbf{d}) \coloneqq \frac{1}{8m} \sum_{j=1}^{m} d_j^2.$$

Then

$$A(\mathbf{z}, \mathbf{d}) \le \frac{1}{m} \sum_{j=1}^{m} [\ell(z_j + d_j, y_j) - \ell(z_j, y_j)] \le A(\mathbf{z}, \mathbf{d}) + Q(\mathbf{d}).$$

Proof. **Step 1: Bound a single loss increment.** For each scalar pair (z_j, y_j), the loss derivative is $\sigma(z_j) - y_j$ and the second derivative belongs to $[0, 1/4]$. Taylor's integral remainder lies in $[0, d_j^2/8]$ for either sign of d_j. Explicitly, the remainder is

$$d_j^2 \int_0^1 (1-u) \sigma'(z_j + u d_j)\, du \in [0, d_j^2/8].$$

Step 2: Average over the evaluation set. Sum the scalar inequalities and divide by m to obtain the stated bounds. □

Finite candidate sets and stopping. When the graph, historical labels, audiences, candidate descriptions, and tie-breaking are fixed, a deterministic recourse-recommendation rule with a unique refit maps a finite set of description configurations into itself. Every trajectory therefore eventually enters a cycle. A round with no edits is a fixed point: it preserves the fitting inputs, predictions, and subsequent recommendations.