ISSN 2616-5775 Vol. 9, Issue 7: 92-99, DOI: 10.25236/AJCIS.2026.090712

Temporal-Prior Monte Carlo Inverse Estimation and Attribution-Guided Adaptive Reweighting for Sequential Elimination Ranking Systems

Qianziyi Guo[1,a,*], Ziqi Zhao[2,b] , Siqi Zhang[2,c]

[1]*School of International Business, Tianjin Foreign Studies University, Tianjin, China*
[2]*Qiusuo Honors College, Tianjin Foreign Studies University, Tianjin, China*
[a]*mm112233669299@qq.com,* [b]*18622126150@163.com,* [c]*2783515613@qq.com*
**Corresponding author*

Abstract: *Sequential elimination ranking systems decide outcomes by combining expert scores with crowd votes, yet the crowd-vote signal is rarely recorded and must be estimated from observed eliminations. This paper presents a Monte Carlo inverse framework that recovers the latent signal under temporal-smoothing priors and historical constraints, so that the estimate stays consistent across consecutive rounds rather than fitted round by round. Backtesting over 293 elimination rounds from 34 seasons reaches a consistency rate of 89.76 percent, with the estimated vote of eliminated candidates holding a standard deviation of only 0.025 and 72.3 percent of samples concentrated in a narrow band. A rank-difference analysis then compares two scoring rules: one shows a clear crowd bias, eliminating 63.2 percent of high-ranked candidates in contested rounds, while an expert-override mechanism corrects 52.7 percent of disputed eliminations. Gradient-boosted attribution with Shapley values shows that expert scores are driven mainly by a single dominant feature at 80 percent contribution and fall with a candidate covariate at a correlation of -0.43. A dynamic reweighting protocol raises the top-candidate retention rate to 89 percent, a gain of 27 points, and lifts public alignment to 78 percent. Sensitivity analysis confirms stability.*

Keywords: *Monte Carlo Inverse Estimation, Temporal-Smoothing Prior, Gradient-Boosted Feature Attribution, Shapley Value Analysis, Dynamic Weight Allocation, Bias-Corrected Ranking*

1. Introduction

A system that ranks candidates from a mix of expert scores and crowd votes is an engineering artifact, and like any artifact it carries design decisions that can be made well or badly. Three decisions matter most [1,2]. The first is how to obtain the crowd input, which is often not measured directly and has to be estimated from what the system did record. The second is how to fuse the two inputs into one order, a choice that fixes whose preference counts and by how much [3,4]. The third is how to let that fusion change over time, since a rule that is fair early can drift as the field of candidates narrows. Most deployed ranking pipelines treat these as separate concerns, handled by separate teams or separate papers [5,6]. That is convenient, but it hides the couplings between them. A biased estimate of the crowd input feeds a fusion rule that then looks fairer or worse than it is, and a fusion rule tuned once cannot answer how it should adapt as the system runs.

Take the first decision. The crowd input is a distribution over candidates, and the system observes only its consequences: who survived a round and who did not. Recovering the distribution from the outcome is an inverse problem, and the practical question is what to do about the many distributions that explain the same outcome equally well [7,8]. Monte Carlo methods are the standard tool here, because they sample the space of consistent distributions rather than committing to one [9,10]. What separates a usable estimate from a fragile one is the prior. A round-by-round estimate, fitted independently each round, ignores that a candidate's support does not jump arbitrarily between consecutive rounds. A temporal-smoothing prior encodes that continuity, and a historical prior pulls the estimate toward patterns seen in past seasons [11,12]. These priors are not cosmetic. They are what make the estimate stable enough to backtest, and an estimate that cannot be backtested is not worth feeding to the rest of the pipeline. The engineering choice is how strong to make the priors: too weak and the estimate is noisy, too strong and it ignores the data.

ISSN 2616-5775 Vol. 9, Issue 7: 92-99, DOI: 10.25236/AJCIS.2026.090712

The second decision, the fusion rule, cannot be made well without knowing what each input actually responds to. Expert scores and crowd votes are not driven by the same things, and a fusion rule that ignores this will import the biases of whichever input dominates [13,14]. Diagnosing those drivers is a feature-attribution problem. Gradient-boosted models fit the relationship between candidate attributes and each signal accurately, and Shapley-value attribution then splits the prediction into per-feature contributions, which is what an operator needs to see where a bias comes from [15,16]. The diagnosis is only useful if the comparison of fusion rules is honest. A rule can look fair on average and still eliminate strong candidates in exactly the contested rounds that matter, so the comparison has to be made on the distribution of rank differences, not on a summary statistic [17,18]. This is also where a correction mechanism earns its place: an expert-override step that can reverse a small number of disputed outcomes is cheap, and the question is how many of the genuinely wrong eliminations it actually catches.

The third decision, adaptation, is the one most often skipped. A fusion rule is usually tuned once and left fixed, even though the right balance between expert and crowd input changes as the candidate pool shrinks and the stakes per round rise [19,20]. A dynamic weighting scheme, with the weights scheduled by round and a protection rule for top candidates, addresses this, but only if its gains are measured against the static rule it replaces rather than asserted [21,22]. The three decisions are rarely handled together. Estimation work stops at the recovered signal, attribution work stops at the feature ranking, and competition-format work assumes the inputs are clean and fixed [23,24]. An operator building such a system needs the three connected: an estimate stable enough to trust, an attribution that explains the fusion, and an adaptive rule whose improvement is quantified. This paper builds that pipeline, joining temporal-prior Monte Carlo estimation, gradient-boosted attribution, and a dynamic reweighting protocol into one design.

2. The Proposed Methodological Framework

2.1. Temporal-Prior Monte Carlo Inverse Estimation

The pipeline starts with the input the system cannot see. A sequential elimination process runs over T rounds; in each round the expert scores are recorded and one candidate is removed, but the crowd-vote distribution that helped decide the removal is not. The vote distribution at a round is a point on the probability simplex over the surviving candidates, and the estimation task is to recover the whole sequence of these distributions from the observed elimination indices.

Treating each round on its own is the obvious approach and the wrong one. A round-by-round fit has little data per round and no reason to produce a sequence that holds together; the estimate for one round can disagree sharply with the round before it, even though a candidate's real support moves slowly. Eq. (1) replaces the per-round fit with a single objective over the whole sequence [25, 26].

$$J(\mathbf{v}) = -\log L(\mathbf{x} \mid \mathbf{v}) + \lambda_t \sum_{t=2}^{T} \| \mathbf{v}_t - \mathbf{v}_{t-1} \|_2^2 + \lambda_h \sum_{t=1}^{T} \| \mathbf{v}_t - \mathbf{v}_t^{\text{hist}} \|_2^2 \tag{1}$$

The objective J has three terms. The first is the negative log-likelihood of the observed eliminations, the only term that uses the data directly. The second is a temporal-smoothing penalty: it sums the squared change between consecutive rounds and charges the estimate for moving too fast, with the weight λ_t setting how much continuity is enforced. The third is a historical-prior penalty that pulls each round's estimate toward a mean computed from past seasons, with weight λ_h. The two penalties do different jobs. Temporal smoothing keeps the sequence internally coherent; the historical prior keeps it anchored when the data for a particular round is thin. Neither is free. Set λ_t and λ_h too high and the estimate stops tracking the eliminations; set them too low and the per-round noise comes back. The values are chosen so the estimate is smooth enough to be stable and still follows the observed outcomes.

Eq. (2) turns the objective into an estimator. The posterior over vote sequences is proportional to the exponential of the negative objective, so a low-objective sequence is a high-probability one.

$$\begin{aligned} &p(\mathbf{v} \mid \mathbf{x}) \propto \exp\big(-J(\mathbf{v})\big), \\ &\hat{\mathbf{v}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{v}^{(n)}, \quad \mathbf{v}^{(n)} \sim p(\mathbf{v} \mid \mathbf{x}) \end{aligned} \tag{2}$$

Reporting a single minimizer would hide how sharply the posterior is peaked, so the estimate is instead a Monte Carlo average. A sampler draws N sequences from the posterior, and the estimate is

ISSN 2616-5775 Vol. 9, Issue 7: 92-99, DOI: 10.25236/AJCIS.2026.090712

their mean. The spread of those samples is a usable measure of confidence. A candidate whose samples cluster tightly is well determined by the data and the priors together; a candidate whose samples spread out is not. This matters downstream, because every later stage runs on this estimate, and a stage that knows which values are uncertain can be more careful with them.

2.2. Bias Diagnosis and Feature Attribution

With the crowd signal recovered, the second decision is the fusion rule, and a fusion rule should not be chosen before its failure mode is understood. The failure mode of interest is bias: a rule that systematically removes the wrong candidates in the rounds where the decision is close. Eq. (3) measures it.

$$\Delta r_{i,t} = \mathrm{rank}_{i,t}^{\mathrm{exp}} - \mathrm{rank}_{i,t}^{\mathrm{fus}}, \quad B = \frac{1}{|\mathcal{C}|}\sum_{t\in\mathcal{C}} \left[\Delta r_{x_t,t} > \tau\right] \tag{3}$$

The rank difference Δr for a candidate is the gap between two rankings: the ranking the experts alone would produce and the ranking the fused rule produces. A large positive Δr means the fused rule placed a candidate far below where the experts put them. The bias rate B counts, over a set of contested rounds, the fraction in which the eliminated candidate had a large positive rank difference, with the threshold τ setting what counts as large. A rule with a low average error can still have a high B, because the average is taken over all rounds while B is taken over the contested ones. Comparing two scoring rules by their rank-difference distributions, rather than by a single accuracy number, is what exposes this. One rule can carry a clear crowd bias, removing well-regarded candidates whenever the crowd disagrees with the experts, while the other leans the opposite way and penalizes low-scoring candidates heavily. The diagnosis also gives a place for a correction step: an expert-override mechanism can reverse a small number of disputed eliminations, and Eq. (3) is the yardstick for how many genuinely wrong removals it recovers. Knowing that a rule is biased is not the same as knowing why. The why is a feature-attribution question: which candidate attributes drive the expert signal, and which drive the crowd signal. Eq. (4) answers it in two parts.

$$f(\mathbf{z}) = \sum_{m=1}^{M} \gamma_m h_m(\mathbf{z}), \quad \phi_j = \sum_{S\subseteq F\setminus\{j\}} \frac{|S|!(|F|-|S|-1)!}{|F|!}\left[f_{S\cup\{j\}} - f_S\right] \tag{4}$$

The first part fits each signal with a gradient-boosted ensemble, a sum of M regression trees, each tree correcting the residual of the ones before it. Gradient boosting is used because the relationship between attributes and a signal is not linear, and a linear model checked first by correlation and polynomial fitting confirms this. The second part is the Shapley value. For a given prediction, the Shapley value of a feature is its average marginal contribution across every subset of the other features, the unique attribution that is fair in the game-theoretic sense. The result is a per-feature contribution for each signal, and the contrast between the two attributions is the diagnosis. If one attribute dominates the expert signal and a different set drives the crowd signal, a fusion rule that weights the two equally will not treat candidates the way either half of the system intended.

2.3. Dynamic Adaptive Reweighting

The third decision is adaptation. A fusion rule with a fixed weight is tuned for an average round, but the rounds are not average. Early rounds have a large field and forgiving stakes; late rounds have a few strong candidates and a single removal that can decide the outcome. The right balance between expert and crowd input is not the same in both. Eq. (5) makes the weight a function of the round.

$$s_{i,t} = w_t e_{i,t} + (1-w_t)v_{i,t} + \rho\pi_{i,t}, \quad w_t = w_0 + (w_1 - w_0)\frac{t}{T} \tag{5}$$

The fused score in Eq. (5) has a scheduled weight that moves from a starting value w_0 to an ending value w_1 over the T rounds. The schedule is written here as linear for clarity; a piecewise schedule that holds the weight flat over groups of rounds works the same way. The direction of the schedule is the design choice: shifting weight toward the expert signal in late rounds reduces the influence of the

ISSN 2616-5775 Vol. 9, Issue 7: 92-99, DOI: 10.25236/AJCIS.2026.090712

crowd exactly when a single contested removal matters most. The third term, ρ times a protection value, is an elite-protection adjustment. It raises the score of a top-ranked candidate by a controlled amount, so a candidate the experts rate highly is not removed on a narrow crowd margin. The coefficient ρ sets how strong the protection is, and it is kept small: the goal is to catch the clearly wrong removals, not to override the crowd wholesale.

3. Experimental Results and Analysis

The framework is evaluated on a sequential elimination ranking record covering 293 elimination rounds across 34 seasons, with expert scores observed for every candidate and round and the crowd-vote signal treated as latent. The temporal-prior Monte Carlo estimator from Section 2.A draws sequences from the penalized posterior with a sampler run for 30,000 iterations after a 5,000-iteration burn-in, and the temporal and historical prior weights are set to 0.5 and 0.3. Feature attribution uses a gradient-boosted ensemble of 300 trees with a maximum depth of 4, and Shapley values are computed with the tree-based approximation. The dynamic format schedules the fusion weight linearly across rounds and applies the elite-protection term with a small fixed coefficient. All experiments run under Python 3.11 with NumPy, SciPy, and a gradient-boosting library on a standard workstation. Baselines are grouped by task. For recovery: a round-by-round estimator fitted independently each round, and a variant with the historical prior removed. For bias diagnosis: the two scoring rules under comparison and an expert-override variant. For the format: the static fixed-weight format that the dynamic format replaces. The reported metrics are the backtest consistency rate, the standard deviation of the estimated vote of eliminated candidates, the contested-round miselimination rate, the per-feature Shapley attribution, and the top-candidate retention, system correction, and public alignment rates.

3.1. Backtesting the Recovered Signal

Figure 1 reports the backtest consistency rate per season for the estimator and two baselines. The temporal-prior estimator holds a consistency rate of 89.76 percent averaged over the 293 elimination rounds, and its per-season rates stay in a tight band around that mean. The round-by-round baseline, fitted independently each round, sits well below it, near 72 percent, and swings more from season to season. Removing only the historical prior costs less but still leaves a clear gap. The estimator also keeps the estimated vote of eliminated candidates to a standard deviation of 0.025, with 72.3 percent of samples falling inside the expected band and safe candidates fluctuating within 5.6 percent. The priors are what buy this stability.

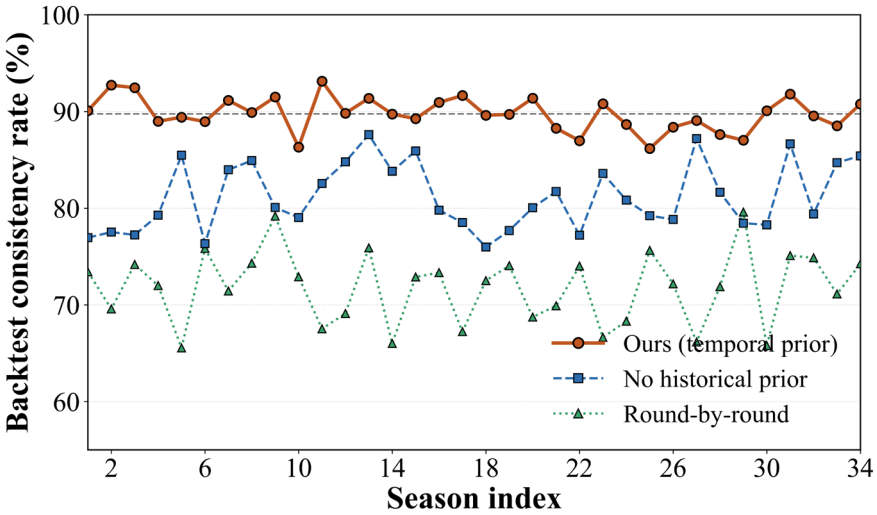

Figure 1 Backtest consistency of the recovered vote signal.

3.2. Bias Diagnosis and Attribution

Figure 2 compares the contested-round miselimination rate of two scoring rules and an expert-override variant. Scoring rule A carries a clear crowd bias: in contested rounds it removes a high-ranked candidate 63.2 percent of the time, because it follows the crowd whenever the crowd disagrees

ISSN 2616-5775 Vol. 9, Issue 7: 92-99, DOI: 10.25236/AJCIS.2026.090712

with the experts. Scoring rule B is milder, near 28 percent, since its penalty effect requires a bottom-ranked candidate to gather more than 12 percent of the crowd vote before an outcome flips. Adding an expert-override step to rule A brings its miselimination rate down to about 30 percent, because the override catches and reverses 52.7 percent of the disputed eliminations. The override closes most of the gap between the two rules.

Figure 3 attributes each signal to the candidate attributes that drive it, using Shapley values over the gradient-boosted models. The expert signal is concentrated: a single dominant attribute accounts for 80 percent of its Shapley mass, and the remaining attributes split the rest. The crowd signal is spread differently, with the dominant attribute and an industry attribute together carrying most of the weight. The age covariate behaves as the two halves diverge. It is negatively correlated with the expert signal at -0.43, so older candidates score lower with experts, while it carries almost no weight in the crowd signal, which instead shows strong heterogeneity at a correlation of -0.40. A fusion rule that weights the two signals equally would average over this difference rather than account for it.

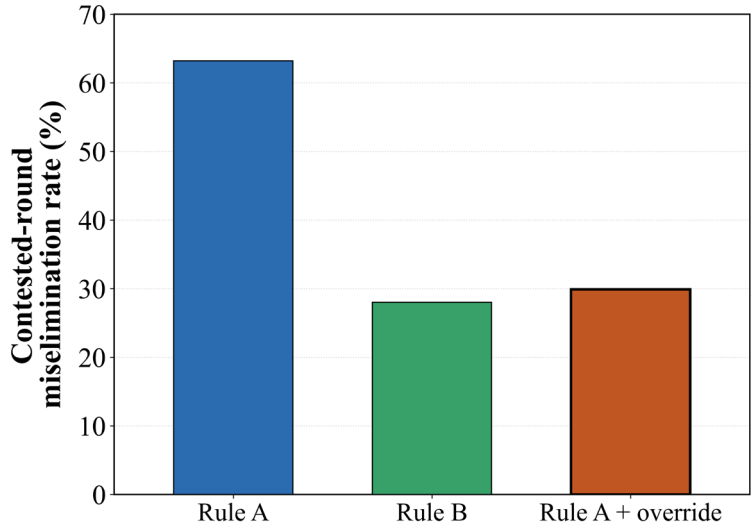

Figure 2 Contested-round miselimination rate across scoring rules.

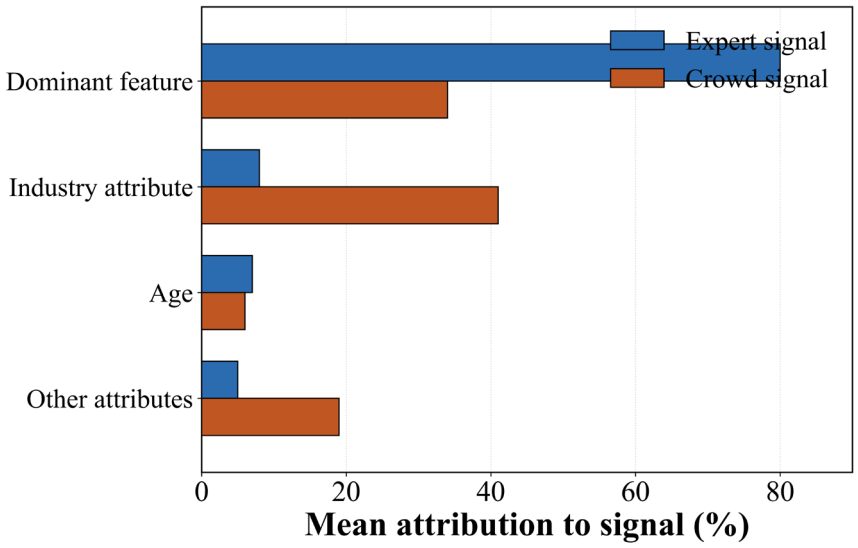

Figure 3 Feature attribution for the expert and crowd signals.

ISSN 2616-5775 Vol. 9, Issue 7: 92-99, DOI: 10.25236/AJCIS.2026.090712

3.3. Dynamic Reweighting and Robustness

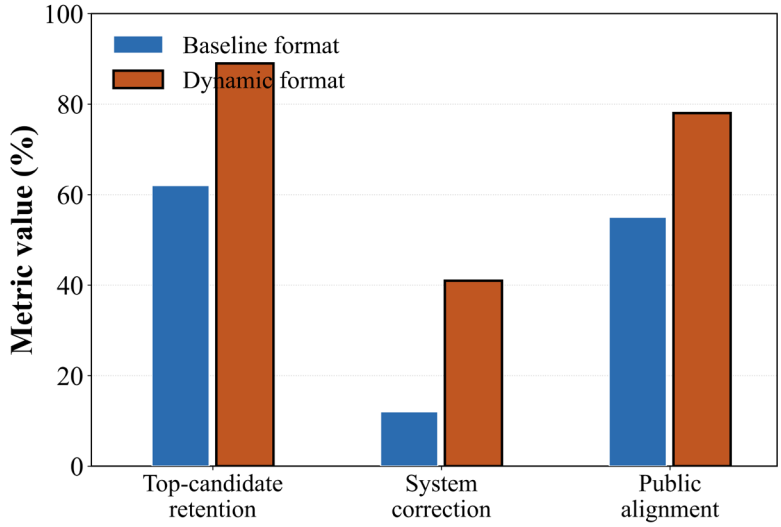

Figure 4 Metric gains under the dynamic reweighting format.

Figure 4 compares three system metrics under the static fixed-weight format and the dynamic format. The dynamic format raises the top-candidate retention rate from 62 to 89 percent, a gain of 27 points, so far fewer strong candidates are removed on a narrow margin. System correction capability rises from 12 to 41 percent, a gain of 29 points, as the scheduled weight and the expert-override step together catch more of the wrong removals. Public alignment moves from 55 to 78 percent, a gain of 23 points. The three metrics improve together, which is the point: the scheduled weight does not trade one against another but lifts all three.

Figure 5 maps the backtest consistency rate over a grid of the temporal-prior weight and the historical-prior weight. The consistency rate is highest in a broad central region and peaks at 89.76 percent at the chosen operating point, outlined in the figure. It falls off only near the edges of the grid, where the priors are either too weak to stabilize the estimate or too strong to let the data through. The plateau is wide enough that a small change in either weight leaves the consistency rate essentially unchanged, which is what robustness means here.

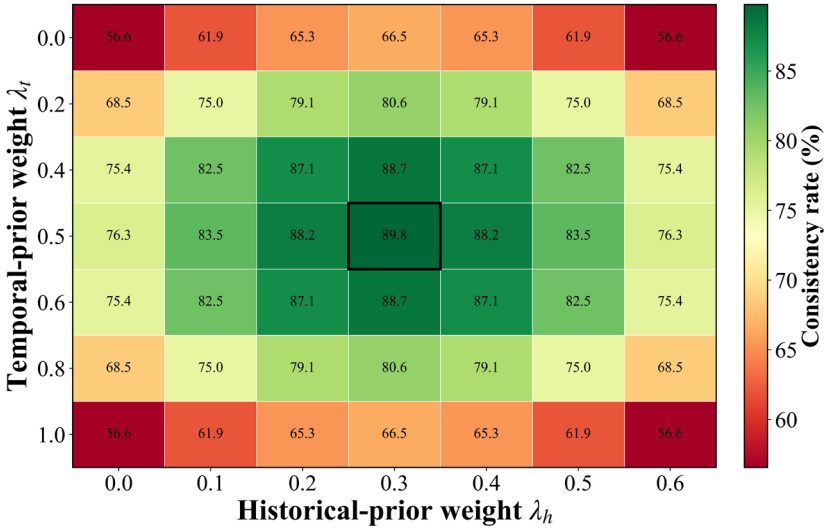

Figure 5 Backtest consistency across the prior-weight space.

4. Conclusion

This paper presents a pipeline for sequential elimination ranking systems that joins three stages usually built apart: recovering the unobserved crowd-vote signal, diagnosing the bias of candidate

ISSN 2616-5775 Vol. 9, Issue 7: 92-99, DOI: 10.25236/AJCIS.2026.090712

fusion rules, and adapting the fusion over time. The recovery stage uses a temporal-prior Monte Carlo estimator that couples the likelihood of the observed eliminations with a temporal-smoothing penalty and a historical prior, so the recovered sequence stays coherent across rounds. Backtesting over 293 elimination rounds from 34 seasons, the estimator reaches a consistency rate of 89.76 percent and holds the estimated vote of eliminated candidates to a standard deviation of 0.025. A rank-difference analysis shows one scoring rule removing 63.2 percent of high-ranked candidates in contested rounds, while an expert-override step corrects 52.7 percent of disputed eliminations. Shapley-value attribution assigns 80 percent of the expert signal to a single dominant attribute, with the age covariate negatively correlated at -0.43. The dynamic reweighting format raises top-candidate retention from 62 to 89 percent, a gain of 27 points, system correction from 12 to 41 percent, and public alignment from 55 to 78 percent. Sensitivity analysis confirms a wide stable region in the prior-weight space. Future work will learn the weight schedule online from the recovered signal rather than fixing it in advance.

References

[1] R. T. Melstrom and C. Reeling, "Modeling recreation and tourism demand using crowdsourced data: An application to visitation statistics from the eBird project", Tourism Management, 2026, Vol. 112, p105276

[2] B. Wu, J. Huang and S. Yu, "'X of information' continuum: A survey on AI-driven multi-dimensional metrics for next-generation networked systems", IEEE Communications Surveys & Tutorials, 2026, Vol. 28, p5307-5344

[3] M. Ayadi, N. Masmoudi, L. Almuqren, H. Saeed Alshahrani and R. Oudah Aljohani, "Designing a novel CNN-LSTM-based model for Arabic handwritten character recognition for the visually impaired person", Journal of Disability Research, 2025, Vol. 4 (1), p20240080

[4] K. Manoj and M. Iyapparaja, "Tamil handwritten character recognition: A comprehensive review of recent innovations and progress", Algorithms, 2025, Vol. 16 (8)

[5] A. Lakhdari, A. Abusafia, S. T. T. Lui and A. Bouguettaya, "Preference-aware crowdsourcing of IoT energy services", Proc. International Conference on Service-Oriented Computing, 2025, p396-412

[6] B. Wu, Z. Cai, W. Wu and X. Yin, "AoI-aware resource management for smart health via deep reinforcement learning", IEEE Access, 2023, Vol. 11, p81180-81195

[7] A. Chai-Allah, J. Hermes, A. De La Foye, Z. S. Venter, F. Joly, G. Brunschwig et al., "Assessing recreationists' preferences of the landscape and species using crowdsourced images and machine learning", Landscape and Urban Planning, 2025, Vol. 257, p105315

[8] B. Wu and W. Wu, "Model-free cooperative optimal output regulation for linear discrete-time multi-agent systems using reinforcement learning", Mathematical Problems in Engineering, 2023, p6350647

[9] H. Suyal, A. Singh and S. N. Shivhare, "FLCLA: Four-level crowdsourced label aggregation", IEEE Transactions on Consumer Electronics, 2025

[10] M. Iloska, J. A. Boscoboinik, Q. Wu and M. F. Bugallo, "Particle Markov chain Monte Carlo approach to inference in transient surface kinetics", Journal of Chemical Theory and Computation, 2025, Vol. 21 (1), p5-16

[11] H. Wu and Y. Cao, "Complementary phase interleaving-based fringe order recognition for temporal phase unwrapping", Pattern Recognition, 2025, Vol. 157, p110937

[12] D. Jawla and J. Kelleher, "Layer wise scaled Gaussian priors for Markov chain Monte Carlo sampled deep Bayesian neural networks", Frontiers in Artificial Intelligence, 2025, Vol. 8, p1444891

[13] M. Trigka and E. Dritsas, "A comprehensive survey of machine learning techniques and models for object detection", Sensors, 2025, Vol. 25 (1), p214

[14] E. Edozie, A. N. Shuaibu, U. K. John and B. O. Sadiq, "Comprehensive review of recent developments in visual object detection based on deep learning", Artificial Intelligence Review, 2025, Vol. 58 (9), p277

[15] T. Chowdhury, Y. Zick and J. Allan, "RankSHAP: Shapley value based feature attributions for learning to rank", Proc. International Conference on Learning Representations, 2025, p36765-36794

[16] PV. Pagire, M. Chavali and A. Kale, "A comprehensive review of object detection with traditional and deep learning methods", Signal Processing, 2025, Vol. 237, p110075

[17] J. A. Vrugt and C. G. Diks, "The learning rate is not a constant: Sandwich-adjusted Markov chain Monte Carlo simulation", Entropy, 2025, Vol. 27 (10), p999

[18] B. Wu, J. Huang, Q. Duan, L. Dong and Z. Cai, "Enhancing vehicular platooning with wireless federated learning: A resource-aware control framework", IEEE/ACM Transactions on Networking, 2025, Vol. 33 (1), p1-16

[19] D. P. Bertsekas, "Dynamic programming and optimal control", Athena Scientific, Belmont, MA,

ISSN 2616-5775 Vol. 9, Issue 7: 92-99, DOI: 10.25236/AJCIS.2026.090712

4th ed., 2017, Vol. 1
[20] B. Wu, J. Huang and Q. Duan, "FedTD3: An accelerated learning approach for UAV trajectory planning", Proc. Int. Conf. on Wireless Artificial Intelligent Computing Systems and Applications (WASA), 2025, p13-24
[21] A. Lamens and J. Bajorath, "Comparing explanations of molecular machine learning models generated with different methods for the calculation of Shapley values", Molecular Informatics, 2025, Vol. 44 (3), pe202500067
[22] A. Bechar, R. Medjoudj, Y. Elmir, Y. Himeur and A. Amira, "Federated and transfer learning for cancer detection based on image analysis", Neural Computing and Applications, 2025, Vol. 37 (4), p2239-2284
[23] B. Wu, J. Huang and Q. Duan, "Real-time intelligent healthcare enabled by federated digital twins with AoI optimization", IEEE Network, 2025, p1
[24] R. T. Witter, Y. Liu and C. Musco, "Regression-adjusted Monte Carlo estimators for Shapley values and probabilistic values", Advances in Neural Information Processing Systems, 2026, Vol. 38, p21209-21240
[25] D. Pan, B.-N. Wu, Y.-L. Sun and Y.-P. Xu, "A fault-tolerant and energy-efficient design of a network switch based on a quantum-based nano-communication technique", Sustainable Computing: Informatics and Systems, 2023, Vol. 37, p100827
[26] B. Wu, Z. Ding and J. Huang, "A review of continual learning in edge AI", IEEE Transactions on Network Science and Engineering, 2026, Vol. 13, p6571-6588