Algorithm 1: DWTS Data Preprocessing

```
Input: Raw CSV file with wide format: each row = one contestant,
       columns = weekly scores
Output: Long format dataset $\mathcal{D}$: each row = (season, week,
        contestant, scores, metrics)
// Load and Parse Data;
Load raw data from CSV file;
for each score column matching weekX_judgeY_score do
    Convert to numeric, set invalid values to NaN;
end
// Process Elimination Information;
for each contestant $c$ do
    Parse results field to extract final_rank and eliminated_week;
    if eliminated_week is missing then
        Infer from last week with valid scores;
    end
    for each week $w >$ eliminated_week do
        Set scores to NaN (post-elimination zeros);
    end
end
// Reshape to Long Format;
for each contestant $c$ do
    for each week $w$ with valid scores do
        Create record: $(c, w, \text{scores}, \sum \text{scores}, \bar{\text{scores}})$;
    end
end
// Compute Competition Metrics;
for each (season, week) group do
    $\texttt{judge\_rank}_i \leftarrow \text{rank}(\text{score}_i)$ using min method;
    $\texttt{judge\_percent}_i \leftarrow \text{score}_i / \sum_j \text{score}_j$;
    contestants_this_week $\leftarrow$ count of contestants;
end
return Processed long-format dataset $\mathcal{D}$;
```

Algorithm 2: Bayesian Fan Vote Share Estimation (Main Controller)

```
Input: Processed dataset $\mathcal{D}$, rule boundaries: $S_{\text{pct}} = 3$, $S_{\text{b2}} = 28$
Output: Fan share estimates with HDI for all (season, week,
        contestant)
Load and normalize data:
 $\texttt{judge\_percent\_norm}_i = \texttt{judge\_percent}_i / \sum_j \texttt{judge\_percent}_j$;
for each (season s, week w) with elimination do
    $\mathcal{W} \leftarrow$ contestants competing in week $w$;
    $e \leftarrow$ identify eliminated contestant;
    // Select model based on season
    if $s < S_{pct}$ then
        $(\text{trace}, \text{contestants}) \leftarrow$ RankBasedModel($\mathcal{W}, e$) ;        // Alg. 3
    else if $s < S_{b2}$ then
        $(\text{trace}, \text{contestants}) \leftarrow$ PercentageModel($\mathcal{W}, e$) ;       // Alg. 4
    else
        $(\text{trace}, \text{contestants}) \leftarrow$ Bottom2Model($\mathcal{W}, e$) ;          // Alg. 5
    end
    // Extract posterior estimates
    for each contestant i do
        $\hat{f}_i \leftarrow \mathbb{E}[\text{fan\_shares}_i]$;
        $\text{HDI}_i \leftarrow$ 95% highest density interval;
        Store $(s, w, i, \hat{f}_i, \text{HDI}_i)$;
    end
end
return All fan share estimates with uncertainty;
```

Algorithm 3: Rank-Based Bayesian Model (Seasons 1–2)

```
Input: Week data $\mathcal{W}$, eliminated contestant $e$, MCMC params:
       $T = 5000$, $B = 2000$, $C = 4$
Output: MCMC trace with posterior samples
// Compute Judge Points from Ranks;
$\mathbf{s} \leftarrow$ judge scores for all $n$ contestants;
$\mathbf{r} \leftarrow \text{rank}(-\mathbf{s})$ ;                    // Highest score → rank 1
$\mathbf{p} \leftarrow n - \mathbf{r} + 1$ ;                           // Rank 1 gets $n$ points
$\tilde{\mathbf{J}} \leftarrow \mathbf{p} / \sum_i p_i$ ;              // Normalize: $\sum_i \tilde{J}_i = 1$
// Build PyMC Model;
$\boldsymbol{\alpha} \leftarrow \mathbf{1}_n$ ;                        // Uniform Dirichlet prior
$\mathbf{f} \sim \text{Dirichlet}(\boldsymbol{\alpha})$ ;             // Fan shares: $\sum_i f_i = 1$
$\mathbf{T} \leftarrow \tilde{\mathbf{J}} + \mathbf{f}$ ;             // Combined: both normalized
// Elimination Constraints;
for each survivor $i \neq e$ do
    $\delta_i \leftarrow T_i - T_e$;
    Add potential: $\log \sigma(\delta_i \cdot \lambda)$ where $\lambda = 100$;
end
// Tie-Breaker Constraints;
for each survivor $i \neq e$ do
    $\tau \leftarrow \sigma(-|T_i - T_e| \cdot 1000 + 5)$ ;                  // Is tie?
    Add potential: $\tau \cdot \log \sigma((f_i - f_e) \cdot \lambda)$;
end
// MCMC Sampling;
Run Slice sampler: $T$ draws, $B$ tuning, $C$ chains;
return trace;
```

Algorithm 4: Percentage-Based Bayesian Model (Seasons 3–27)

```
Input: Week data $\mathcal{W}$, eliminated contestant $e$, MCMC params:
       $T = 2000$, $B = 1000$, $C = 2$
Output: MCMC trace with posterior samples
// Compute Normalized Judge Percentage;
$\mathbf{s} \leftarrow$ judge scores for all $n$ contestants;
$\tilde{\mathbf{J}} \leftarrow \mathbf{s}/\sum_i s_i$ ;                    // Normalize: $\sum_i \tilde{J}_i = 1$
// Build PyMC Model;
$\boldsymbol{\alpha} \leftarrow \mathbf{1}_n$ ;                        // Uniform Dirichlet prior
$\mathbf{f} \sim \text{Dirichlet}(\boldsymbol{\alpha})$ ;                 // Fan shares: $\sum_i f_i = 1$
$\mathbf{T} \leftarrow \tilde{\mathbf{J}} + \mathbf{f}$ ;                   // Combined: both normalized
// Elimination Constraint (Lowest Score Eliminated);
for each survivor $i \neq e$ do
    $\delta_i \leftarrow T_i - T_e$;
    Add potential: $\log \sigma(\delta_i \cdot \lambda)$ where $\lambda = 100$;
end
// MCMC Sampling;
Run NUTS sampler: $T$ draws, $B$ tuning, $C$ chains;
return trace;
```

Algorithm 5: Bottom-2 Rule Bayesian Model (Seasons 28+)

```
Input: Week data $\mathcal{W}$, eliminated contestant $e$, MCMC params:
       $T = 3000$, $B = 1500$, $C = 4$
Output: MCMC trace with posterior samples
// Compute Judge Points (Same as Rank-Based);
$\mathbf{s} \leftarrow$ judge scores, $\mathbf{r} \leftarrow \text{rank}(-\mathbf{s})$;
$\mathbf{p} \leftarrow n - \mathbf{r} + 1$, $\tilde{\mathbf{J}} \leftarrow \mathbf{p} / \sum_i p_i$ ;          // $\sum_i \tilde{J}_i = 1$
// Build PyMC Model with Informative Prior;
$\mathbf{q} \leftarrow (\text{rank}(\mathbf{s}) - 1)/(n - 1)$ ;          // Judge percentile $\in [0, 1]$
$\boldsymbol{\alpha} \leftarrow 0.5 + \kappa \cdot \mathbf{q}$ where $\kappa = 5$ ;          // Informative prior
$\mathbf{f} \sim \text{Dirichlet}(\boldsymbol{\alpha})$ ;          // $\sum_i f_i = 1$
$\mathbf{T} \leftarrow \tilde{\mathbf{J}} + \mathbf{f}$ ;          // Combined: both normalized
// Soft Bottom-2 Constraint;
$N_{\text{lower}} \leftarrow 0$;
for each survivor $i \neq e$ do
    $\delta_i \leftarrow T_e - T_i$ ;          // Positive if $e$ higher than $i$
    $N_{\text{lower}} \leftarrow N_{\text{lower}} + \sigma(\delta_i \cdot \gamma)$ where $\gamma = 20$;
end
excess $\leftarrow N_{\text{lower}} - \theta$ where $\theta = 1.3$;
Add potential: $\begin{cases} -\rho \cdot \text{excess}^2 & \text{if excess} > 0 \\ 0 & \text{otherwise} \end{cases}$ where $\rho = 20$;
// MCMC Sampling;
Run Slice sampler: $T$ draws, $B$ tuning, $C$ chains;
return trace;
```

Algorithm 6: Elimination Consistency Validation

```
Input: Fan share estimates $\hat{\mathbf{f}}$, original data $\mathcal{D}$
Output: Validation results: consistency rate, margin distribution
for each (season $s$, week $w$) do
    rule $\leftarrow$ determine rule type based on $s$;
    Compute judge component $\mathbf{J}$ based on rule;
    $\mathbf{T} \leftarrow \mathbf{J} + \hat{\mathbf{f}}$ ;                    // Reconstructed total scores
    if rule $\in$ {rank, percentage} then
        consistent $\leftarrow (T_e \leq \min_{i \neq e} T_i)$;
        margin $\leftarrow \min_{i \neq e} T_i - T_e$;
    else
        $N_{\text{lower}} \leftarrow |\{i : T_i < T_e, i \neq e\}|$;
        consistent $\leftarrow (N_{\text{lower}} \leq 1)$ ;                    // In Bottom 2
        margin $\leftarrow 1 - N_{\text{lower}}$;
    end
    Store $(s, w, \text{rule}, \text{consistent}, \text{margin}, \hat{f}_e)$;
end
// Aggregate Statistics;
for each rule type do
    Compute: $n$, consistency rate, mean margin ± std;
end
return Validation DataFrame;
```

Algorithm 7: Fan Vote Uncertainty Analysis

```
Input: Fan share estimates with HDI, MCMC diagnostics
Output: Uncertainty metrics and visualizations
// Compute HDI Width;
for each (season, week, contestant) do
    HDI_width ← HDI_upper − HDI_lower;
end
// Analyze by Rule Type;
for each rule type ∈ {rank, percentage, bottom2} do
    Compute mean HDI width, distribution;
end
// Contestant Count vs Uncertainty;
for each (season, week) do
    $n$ ← number of contestants;
    $\bar{w}$ ← mean HDI width for week;
end
Fit linear regression: $\bar{w} = \beta_0 + \beta_1 n$;
Compute correlation $r$;
// MCMC Convergence Check (S1-S2);
for each (season, week) in S1-S2 do
    Check: $\hat{R}_{\max} < 1.05$, $\mathrm{ESS}_{\min} > 400$;
    Flag non-converged weeks;
end
// Generate Visualizations;
Plot (a): HDI width distribution by era (violin + box);
Plot (b): Uncertainty vs contestant count (scatter + regression);
Plot (c): Eliminated vs survived uncertainty (box);
Plot (d): MCMC convergence (ESS bars + R-hat line);
return Uncertainty analysis figures;
```

Algorithm 8: Scoring Method Comparison (Rank vs Percentage)

```
Input: Fan share estimates $\hat{\mathbf{f}}$, original data $\mathcal{D}$, seasons 1 to 27
Output: Rule comparison metrics: fan share changes under alternative
        rules
for each (season $s$, week $w$) with elimination do
    $\mathcal{W} \leftarrow$ contestants in week $w$;
    $e \leftarrow$ eliminated contestant;
    $\mathbf{s} \leftarrow$ judge scores;
    // Compute judge components under both rules
    $\mathbf{r} \leftarrow \text{rank}(-\mathbf{s}), \quad \mathbf{p} \leftarrow n - \mathbf{r} + 1$;
    $\mathbf{J}_{\text{rank}} \leftarrow \mathbf{p} / \sum_i p_i$;
    $\mathbf{J}_{\text{pct}} \leftarrow \mathbf{s} / \sum_i s_i$;
    // Determine actual rule and compute counterfactual
    $\text{actual} \leftarrow \begin{cases} \text{rank} & s \leq 2 \\ \text{percentage} & s > 2 \end{cases}$;
    // Compute required fan share under counterfactual rule
    $\mathbf{J}_{\text{cf}} \leftarrow \begin{cases} \mathbf{J}_{\text{pct}} & \text{actual} = \text{rank} \\ \mathbf{J}_{\text{rank}} & \text{otherwise} \end{cases}$;
    $\Delta J_e \leftarrow \min_{i \neq e} J_{\text{cf},i} - J_{\text{cf},e}$;
    $f_{\text{critical}} \leftarrow \Delta J_e \cdot \frac{n-1}{n} + \frac{1}{n}$;
    Store $(s, w, \hat{f}_e, f_{\text{critical}}, \text{actual})$;
end
// Aggregate by Rule Type;
for each rule $\in$ {rank, percentage} do
    $\bar{f}_{\text{elim}} \leftarrow$ mean eliminated fan share under this rule;
    $\bar{f}_{\text{cf}} \leftarrow$ mean counterfactual critical share;
end
Conclusion $\leftarrow (\bar{f}_{\text{elim}}^{\text{pct}} > \bar{f}_{\text{elim}}^{\text{rank}})$ implies percentage favors fans;
return Rule comparison DataFrame;
```

Algorithm 9: Controversial Contestant Identification

```
Input: Original data $\mathcal{D}$, threshold $K = 15$
Output: Top $K$ controversial contestants with controversy scores
for each contestant $c$ in $\mathcal{D}$ do
    $R_{\text{final}} \leftarrow$ final placement rank of $c$;
    $W_{\text{total}} \leftarrow$ total weeks competed;
    $W_{\text{bottom}} \leftarrow 0, \quad W_{\text{bottom2}} \leftarrow 0$;
    for each week $w$ contestant $c$ participated do
        $\mathbf{s}_w \leftarrow$ all judge scores in week $w$;
        $r_c \leftarrow \text{rank}(s_c)$ using min method (1 = lowest);
        if $r_c = 1$ then
            $W_{\text{bottom}} \leftarrow W_{\text{bottom}} + 1$;
        end
        if $r_c \leq 2$ then
            $W_{\text{bottom2}} \leftarrow W_{\text{bottom2}} + 1$;
        end
    end
    // Controversy score:  low scores but high placement
    if $R_{final} \leq 5$ then
        $\text{score}_c \leftarrow \frac{W_{\text{bottom}}}{R_{\text{final}}} + \frac{W_{\text{bottom2}}}{2 \cdot R_{\text{final}}}$;
    else
        $\text{score}_c \leftarrow \frac{W_{\text{bottom}}}{2 \cdot R_{\text{final}}}$;
    end
end
Sort contestants by $\text{score}_c$ descending;
return Top $K$ contestants;
```

Algorithm 10: Monte Carlo Counterfactual Survival Simulation

```
Input: Week data $\mathcal{W}$, fan estimates $(\hat{f}_i, \sigma_i)$, target $t$, rule $\in$ {rank,
       pct}, $M = 10000$
Output: Survival probability $P_{\text{survive}}(t)$
$\mathbf{s} \leftarrow$ judge scores for all $n$ contestants;
// Compute judge component based on rule
if rule = rank then
    $\mathbf{r} \leftarrow \text{rank}(-\mathbf{s}), \quad \mathbf{J} \leftarrow (n - \mathbf{r} + 1)/\sum_i (n - r_i + 1)$;
else
    $\mathbf{J} \leftarrow \mathbf{s}/\sum_i s_i$;
end
// Monte Carlo simulation
$N_{\text{survive}} \leftarrow 0$;
for $m = 1$ to $M$ do
    // Sample fan shares from posterior
    for each contestant $i$ do
        $\tilde{f}_i^{(m)} \sim \mathcal{N}(\hat{f}_i, \sigma_i^2)$;
        $\tilde{f}_i^{(m)} \leftarrow \text{clip}(\tilde{f}_i^{(m)}, 0.001, 0.999)$;
    end
    $\tilde{\mathbf{f}}^{(m)} \leftarrow \tilde{\mathbf{f}}^{(m)}/\sum_i \tilde{f}_i^{(m)}$ ;                // Normalize
    $\mathbf{T}^{(m)} \leftarrow 0.5 \cdot \mathbf{J} + 0.5 \cdot \tilde{\mathbf{f}}^{(m)}$;
    $e^{(m)} \leftarrow \arg\min_i T_i^{(m)}$;
    if $e^{(m)} \neq t$ then
        $N_{\text{survive}} \leftarrow N_{\text{survive}} + 1$;
    end
end
return $P_{\text{survive}} = N_{\text{survive}}/M$;
```

Algorithm 11: Critical Fan Share Analysis with Bottom-2 Rule

```
Input: Week data $\mathcal{W}$, target contestant $t$
Output: Critical fan shares $(f^*_{\text{rank}}, f^*_{\text{pct}}, f^*_{\text{B2}})$, risk assessment
$\mathbf{s} \leftarrow$ judge scores, $n \leftarrow |\mathcal{W}|$;
$\mathbf{J}_{\text{rank}} \leftarrow$ rank-based normalized points;
$\mathbf{J}_{\text{pct}} \leftarrow \mathbf{s} / \sum_i s_i$;
// Critical share to avoid last place (Rank/Pct rules)
Function CriticalNotLast($\mathbf{J}$, $t$, $n$):
    $\Delta J \leftarrow \min_{i \neq t} J_i - J_t$;
    return $\max\left(0, \min\left(1, \Delta J \cdot \frac{n-1}{n} + \frac{1}{n}\right)\right)$;
$f^*_{\text{rank}} \leftarrow$ CriticalNotLast($\mathbf{J}_{rank}$, $t$, $n$);
$f^*_{\text{pct}} \leftarrow$ CriticalNotLast($\mathbf{J}_{pct}$, $t$, $n$);
// Critical share under Bottom-2 rule
$r_t \leftarrow$ score rank of $t$ (1 = lowest);
$N_{\text{lower}} \leftarrow |\{i : s_i < s_t\}|$;
if $N_{lower} = 0$ ;                                  // $t$ has lowest score
 then
    // Must avoid bottom 2; judges will eliminate
    $J_{\text{2nd}} \leftarrow$ judge component of 2nd lowest scorer;
    $f^*_{\text{B2}} \leftarrow \max($CriticalNotLast($\mathbf{J}_{pct}$, $t$, $n$)$, 0.5)$;
    risk $\leftarrow$ HIGH,    judge_decision $\leftarrow$ ELIMINATE;
else if $N_{lower} = 1$ ;                                  // $t$ is 2nd lowest
 then
    // If in bottom 2, judges save $t$ (not lowest)
    $f^*_{\text{B2}} \leftarrow$ CriticalNotLast($\mathbf{J}_{pct}$, $t$, $n$);
    risk $\leftarrow$ MEDIUM,    judge_decision $\leftarrow$ SAVE;
else
    // Not in bottom 2 risk zone
    $f^*_{\text{B2}} \leftarrow$ CriticalNotLast($\mathbf{J}_{pct}$, $t$, $n$);
    risk $\leftarrow$ LOW,    judge_decision $\leftarrow$ N/A;
end
// Compute rule impact gaps
$\Delta_{\text{pct-rank}} \leftarrow f^*_{\text{pct}} - f^*_{\text{rank}}$;
$\Delta_{\text{B2-pct}} \leftarrow f^*_{\text{B2}} - f^*_{\text{pct}}$;
return $(f^*_{\text{rank}}, f^*_{\text{pct}}, f^*_{\text{B2}}, \text{risk}, \text{judge\_decision}, \Delta_{\text{pct-rank}}, \Delta_{\text{B2-pct}})$;
```

Algorithm 12: Counterfactual Survival Heatmap Generation

```
Input: Controversial contestants $\mathcal{C}$, survival simulations, original data $\mathcal{D}$
Output: Heatmap visualization of counterfactual survival probabilities
// Prepare Heatmap Data;
for each contestant $c \in \mathcal{C}$ do
    $s \leftarrow$ season of $c$;
    $\text{actual\_rule} \leftarrow \begin{cases} \text{rank} & s \leq 2 \\ \text{bottom2} & s \geq 28 \\ \text{percentage} & \text{otherwise} \end{cases}$;
    for each week $w$ contestant $c$ participated do
        $\tilde{s}_w \leftarrow (s_c - s_{\min})/(s_{\max} - s_{\min})$ ;        // Normalized score
        // Counterfactual rule (opposite of actual)
        $P_{\text{cf}} \leftarrow \begin{cases} P^{\text{pct}}_{\text{survive}} & \text{actual} = \text{rank} \\ P^{\text{rank}}_{\text{survive}} & \text{otherwise} \end{cases}$;
        Add row: $(c, w, \tilde{s}_w, P_{\text{cf}}, \text{actual\_rule})$;
    end
end

// Generate Heatmap;
Sort contestants by final rank (winners first);
$X \leftarrow$ week numbers,    $Y \leftarrow$ contestant labels;
$Z \leftarrow$ matrix of $P_{\text{cf}}$ values;
Apply diverging colormap: red ($P < 0.5$) to green ($P > 0.5$);
Overlay normalized judge score as marker size;
Add annotations for actual rule era;
return Heatmap figure;
```

Algorithm 13: Linear Mixed-Effects Model for Celebrity & Partner Effects

```
Input: Dataset $\mathcal{D}$ with outcomes $y^{(J)}$ (judge score), $y^{(F)}$ (log fan
       share), fixed effects $\mathbf{X}$, grouping factor $g$ (partner)
Output: Fixed effects $\hat{\boldsymbol{\beta}}$, random effects $\hat{\mathbf{u}}$, ICC
// Model Specification;
Fixed effects: $\mathbf{X} = [\text{Age}, \text{Industry}_1, \ldots, \text{Industry}_k, \text{Week}, \text{StageRatio}]$;
Random intercept: $u_g \sim \mathcal{N}(0, \sigma_u^2)$ for each partner $g$;
Model: $y_{ig} = \mathbf{x}_i^\top \boldsymbol{\beta} + u_g + \varepsilon_{ig}, \quad \varepsilon_{ig} \sim \mathcal{N}(0, \sigma^2)$;
// REML Estimation;
Construct marginal covariance: $\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}^\top + \sigma^2\mathbf{I}$;
Maximize restricted log-likelihood:;
  $\ell_R = -\frac{1}{2}\Big[\log|\mathbf{V}| + \log|\mathbf{X}^\top\mathbf{V}^{-1}\mathbf{X}| + (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})\Big]$;
Estimate $\hat{\sigma}_u^2$, $\hat{\sigma}^2$ via Powell optimizer;
// Extract Partner Random Effects (BLUPs);
$\hat{\mathbf{u}} = \mathbf{G}\mathbf{Z}^\top\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$;
// Compute Intraclass Correlation (ICC);
$\rho = \frac{\hat{\sigma}_u^2}{\hat{\sigma}_u^2 + \hat{\sigma}^2}$ ;                    // Variance explained by partner
// Fit Both Models;
$(\hat{\boldsymbol{\beta}}^{(J)}, \hat{\mathbf{u}}^{(J)}, \rho^{(J)}) \leftarrow$ fit LMM on $y^{(J)}$;
$(\hat{\boldsymbol{\beta}}^{(F)}, \hat{\mathbf{u}}^{(F)}, \rho^{(F)}) \leftarrow$ fit LMM on $y^{(F)}$;
return $\hat{\boldsymbol{\beta}}^{(J)}$, $\hat{\boldsymbol{\beta}}^{(F)}$, $\{\hat{u}_g\}$, ICC values;
```

Algorithm 14: XGBoost Feature Importance with SHAP Decomposition

```
Input: Feature matrix $\mathbf{X} \in \mathbb{R}^{N\times p}$, targets $y^{(J)}$, $y^{(F)}$, params: $T = 100$
       trees, $d = 4$ depth, $\eta = 0.1$
Output: Feature importance rankings, SHAP values for both models
// Feature Engineering;
$\mathbf{X} \leftarrow$
  $[\mathrm{Age}, \mathrm{Week}, \mathrm{StageRatio}, \mathrm{Ind}_1, \dots, \mathrm{Ind}_k, \mathrm{Age}_1, \dots, \mathrm{Age}_m, \mathrm{Partner}_{\mathrm{enc}}]$;
$y^{(F)} \leftarrow \log(y^{(F)} + 10^{-6})$ ;                    // Log-transform fan share
// Train XGBoost Models;
for target $y \in \{y^{(J)}, y^{(F)}\}$ do
    Initialize $\hat{y}^{(0)} = 0$;
    for $t = 1$ to $T$ do
        $g_i = \frac{\partial \ell(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}, \quad h_i = \frac{\partial^2 \ell}{\partial (\hat{y}_i^{(t-1)})^2}$;
        Fit tree $f_t$ minimizing $\sum_i [g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$;
        $\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta \cdot f_t(\mathbf{x})$;
    end
end
// SHAP Value Computation (TreeExplainer);
for each model $\mathcal{M} \in \{\mathcal{M}_J, \mathcal{M}_F\}$ do
    for each sample $\mathbf{x}_i$ and feature $j$ do
        $\phi_j(\mathbf{x}_i) = \sum_{S \subseteq \mathcal{F} \setminus \{j\}} \frac{|S|!(p-|S|-1)!}{p!} [f_{\mathcal{M}}(S \cup \{j\}) - f_{\mathcal{M}}(S)]$;
    end
end
// Feature Importance Ranking;
for each feature $j$ do
    $I_j = \frac{1}{N} \sum_{i=1}^{N} |\phi_j(\mathbf{x}_i)|$ ;                    // Mean absolute SHAP
end
return Rankings $\mathcal{R}^{(J)}$, $\mathcal{R}^{(F)}$, SHAP matrices $\Phi^{(J)}$, $\Phi^{(F)}$;
```

Algorithm 15: Bootstrap Test for Judge vs Fan Effect Differences

```
Input: Dataset $\mathcal{D}$, feature $f$, bootstrap iterations $B = 1000$
Output: Effect estimates, confidence intervals, p-values
// Standardize Outcomes;
$y^{(J)} \leftarrow$ standardized judge score (z-score);
$y^{(F)} \leftarrow$ standardized log fan share;
// Bootstrap Resampling;
$\mathcal{B}_{\text{diff}} \leftarrow \emptyset$;
for $b = 1$ to $B$ do
    Sample $\mathcal{D}^{(b)}$ with replacement from $\mathcal{D}$;
    Fit $\hat{\beta}^{(J,b)} \leftarrow$ OLS of $y^{(J)}$ on $f$ using $\mathcal{D}^{(b)}$;
    Fit $\hat{\beta}^{(F,b)} \leftarrow$ OLS of $y^{(F)}$ on $f$ using $\mathcal{D}^{(b)}$;
    $\mathcal{B}_{\text{diff}} \leftarrow \mathcal{B}_{\text{diff}} \cup \{\hat{\beta}^{(J,b)} - \hat{\beta}^{(F,b)}\}$;
end
// Original Estimates;
$\hat{\beta}^{(J)} \leftarrow$ OLS on full $\mathcal{D}$,    $\hat{\beta}^{(F)} \leftarrow$ OLS on full $\mathcal{D}$;
$\hat{\Delta} \leftarrow \hat{\beta}^{(J)} - \hat{\beta}^{(F)}$;
// Confidence Interval & p-value;
$\text{CI}_{95\%} \leftarrow [\text{percentile}(\mathcal{B}_{\text{diff}}, 2.5), \text{percentile}(\mathcal{B}_{\text{diff}}, 97.5)]$;
$p \leftarrow 2 \cdot \min\left(\frac{|\{d \in \mathcal{B}_{\text{diff}} : d > 0\}|}{B}, \frac{|\{d \in \mathcal{B}_{\text{diff}} : d < 0\}|}{B}\right)$;
significant $\leftarrow (\text{CI}_{\text{lower}} > 0) \vee (\text{CI}_{\text{upper}} < 0)$;
return $(\hat{\beta}^{(J)}, \hat{\beta}^{(F)}, \hat{\Delta}, \text{CI}_{95\%}, p, \text{significant})$;
```

Algorithm 16: Age × Industry Interaction Effect Visualization

```
Input: Dataset $\mathcal{D}$ with age, industry, $y^{(J)}$, $y^{(F)}$
Output: Interaction heatmaps for judges and fans
// Create Age Bins;
$\text{AgeBin} \leftarrow \begin{cases} \text{“¡25”} & \text{if age} < 25 \\ \text{“25-35”} & \text{if } 25 \leq \text{age} < 35 \\ \text{“35-45”} & \text{if } 35 \leq \text{age} < 45 \\ \text{“45-55”} & \text{if } 45 \leq \text{age} < 55 \\ \text{“55+”} & \text{if age} \geq 55 \end{cases}$;
// Compute Cell Means;
for each industry $I \in \mathcal{I}$ do
    for each age bin $A \in \mathcal{A}$ do
        $\mu^{(J)}_{I,A} \leftarrow \text{mean}(y^{(J)} | \text{Industry} = I, \text{AgeBin} = A)$;
        $\mu^{(F)}_{I,A} \leftarrow \text{mean}(y^{(F)} | \text{Industry} = I, \text{AgeBin} = A)$;
        $n_{I,A} \leftarrow$ count of samples in cell;
    end
end
// Standardize Fan Share Matrix;
$\tilde{\mu}^{(F)} \leftarrow \frac{\mu^{(F)} - \bar{\mu}^{(F)}}{\text{std}(\mu^{(F)})}$;
// Compute Difference Matrix;
$\Delta_{I,A} \leftarrow \mu^{(J)}_{I,A} - \tilde{\mu}^{(F)}_{I,A}$;
// Generate Heatmaps;
Plot Judge Score heatmap: $\mathbf{M}^{(J)}$ with RdYlGn colormap;
Plot Fan Vote heatmap: $\tilde{\mathbf{M}}^{(F)}$ with RdYlGn colormap;
Plot Difference heatmap: $\boldsymbol{\Delta}$ with PuOr colormap;
Annotate cells with values and sample sizes;
return Heatmap figures;
```

Algorithm 17: Cox Proportional Hazards Model for Elimination Risk

```
Input: Survival data: time $T_i$ (weeks survived), event $\delta_i$
       (1=eliminated, 0=winner), covariates $\mathbf{x}_i$
Output: Hazard ratios, survival curves, partner effects
// Prepare Survival Data;
for each contestant $c$ in season $s$ do
    $T_c \leftarrow$ eliminated_week (or max week if winner);
    $\delta_c \leftarrow \mathbf{1}[\text{final\_rank} > 1]$ ;          // 1=eliminated, 0=censored
    $\mathbf{x}_c \leftarrow [\text{Age}, \text{Ind}_1, \dots, \text{Ind}_k, \text{Age}_{\text{U25}}, \text{Age}_{40-55}, \text{Age}_{55+}]$;
end
// Cox Model: Partial Likelihood Estimation;
Hazard function: $h(t|\mathbf{x}) = h_0(t)\exp(\boldsymbol{\beta}^\top \mathbf{x})$;
Partial log-likelihood:;
  $\ell(\boldsymbol{\beta}) = \sum_{i:\delta_i=1} \left[\boldsymbol{\beta}^\top \mathbf{x}_i - \log \sum_{j \in \mathcal{R}(t_i)} \exp(\boldsymbol{\beta}^\top \mathbf{x}_j)\right]$;
where $\mathcal{R}(t)$ is risk set at time $t$;
Maximize $\ell(\boldsymbol{\beta})$ via Newton-Raphson;
// Compute Hazard Ratios;
for each covariate $k$ do
    $\text{HR}_k = \exp(\hat{\beta}_k)$;
    $\text{CI}_{95\%} = [\exp(\hat{\beta}_k - 1.96 \cdot \text{SE}_k), \exp(\hat{\beta}_k + 1.96 \cdot \text{SE}_k)]$;
    if $HR_k > 1$ then Higher elimination risk;
    else Lower elimination risk (protective);
end
// Kaplan-Meier Survival Curves by Group;
for each group $G \in \{Industries, AgeBrackets\}$ do
    $\hat{S}_G(t) = \prod_{t_i \le t} \left(1 - \frac{d_i}{n_i}\right)$ ;          // K-M estimator
    where $d_i$ = events at $t_i$, $n_i$ = at risk at $t_i$;
end
// Pro Partner Survival Analysis;
for each partner $g$ with $n_g \ge 5$ do
    $\tilde{T}_g \leftarrow$ median survival time;
    $\text{WinRate}_g \leftarrow \frac{|\{c:\text{rank}_c=1, \text{partner}_c=g\}|}{n_g}$;
    $\text{Top3Rate}_g \leftarrow \frac{|\{c:\text{rank}_c \le 3, \text{partner}_c=g\}|}{n_g}$;
end
return Hazard ratios, K-M curves, partner survival statistics;
```

Algorithm 18: Cross-Validation of LMM, XGBoost, and Cox Models

```
Input: Results from Algorithms 13, 14, 17
Output: Consistency assessment across models
// Extract Effect Directions;
for each feature $f \in \{Age, Industries, AgeBrackets\}$ do
    $\mathrm{sign}^{(J)}_{\mathrm{LMM}} \leftarrow \mathrm{sign}(\hat{\beta}^{(J)}_f)$ ;                    // LMM judge effect
    $\mathrm{sign}^{(F)}_{\mathrm{LMM}} \leftarrow \mathrm{sign}(\hat{\beta}^{(F)}_f)$ ;                      // LMM fan effect
    $\mathrm{sign}_{\mathrm{XGB}} \leftarrow \mathrm{sign}(\bar{\phi}_f)$ ;                  // Mean SHAP direction
    $\mathrm{sign}_{\mathrm{Cox}} \leftarrow \mathrm{sign}(\log \mathrm{HR}_f)$ ;                  // Cox risk direction
end
// Consistency Check;
for each feature $f$ do
    $\mathrm{consistent}_f \leftarrow (\mathrm{sign}^{(J)}_{\mathrm{LMM}} = \mathrm{sign}_{\mathrm{XGB}}) \wedge (\mathrm{sign}^{(J)}_{\mathrm{LMM}} = -\mathrm{sign}_{\mathrm{Cox}})$;
    // Higher score ⇒ lower elimination risk
end
// Partner Effect Consistency;
Compute Spearman correlation:;
$r_{\text{LMM-Cox}} = \mathrm{corr}(\hat{u}^{\mathrm{LMM}}_g, \mathrm{Top3Rate}^{\mathrm{Cox}}_g)$;
// Summary Table;
for each feature do
    Report: LMM $\hat{\beta}$, XGBoost SHAP rank, Cox HR, consistency flag;
end
return Validation report with consistency metrics;
```

Algorithm 19: Multi-Objective Optimization with Pareto Frontier

```
Input: Merged data $\mathcal{D}$ with judge scores and fan shares, weight grid
       $\mathbf{w} = [0, 0.05, \dots, 1]$
Output: Pareto frontier, objective values for each weight
// Define Objective Functions;
for each weight $w \in \mathbf{w}$ do
    for each (season, week) with $|\mathcal{W}| \geq 3$ do
        // Simulate elimination
        $\tilde{s}_i \leftarrow$ min-max normalized judge score;
        $\tilde{f}_i \leftarrow$ normalized fan share ($\sum_i \tilde{f}_i = 1$);
        $T_i \leftarrow w \cdot \tilde{s}_i + (1 - w) \cdot \tilde{f}_i$;
        $e \leftarrow \arg\min_i T_i$ ;                    // Eliminated contestant
        $r_e \leftarrow$ judge rank of eliminated (1 = highest);
        // Track metrics
        $\text{fairness}_w \leftarrow r_e/n$ ;       // Higher = eliminated low scorer
        $\text{robbery}_w \leftarrow \mathbf{1}[r_e \leq 2]$ ;               // Top-2 eliminated
        $\text{decisive}_w \leftarrow \mathbf{1}[e \neq e_{w=1}]$ ;           // Fan changed outcome
    end
    // Aggregate objectives
    $F_w \leftarrow \overline{\text{fairness}}_w$;
    $E_w \leftarrow \overline{\text{decisive}}_w$ ;                                // Engagement
    $R_w \leftarrow 1 - \overline{\text{robbery}}_w$ ;                         // No-Robbery rate
end
// Construct Pareto Frontier;
Sort solutions by engagement descending;
$\mathcal{P} \leftarrow \{(E_w, F_w, R_w) : w \in \mathbf{w}\}$;
return Pareto frontier $\mathcal{P}$, objective values;
```

Algorithm 20: Knee Point Detection for Optimal Weight Selection

```
Input: Pareto frontier $\mathcal{P} = \{(E_i, F_i)\}_{i=1}^{n}$, corresponding weights $\mathbf{w}$
Output: Knee points $w^*_{\text{Kneedle}}$, $w^*_{\text{marginal}}$
// Method 1: Kneedle Algorithm;
// Normalize to $[0, 1]$
$\tilde{E}_i \leftarrow (E_i - E_{\min})/(E_{\max} - E_{\min})$;
$\tilde{F}_i \leftarrow (F_i - F_{\min})/(F_{\max} - F_{\min})$;
// Line from first to last point:  $ax + by + c = 0$
$a \leftarrow \tilde{F}_n - \tilde{F}_1, \quad b \leftarrow -(\tilde{E}_n - \tilde{E}_1)$;
$c \leftarrow (\tilde{E}_n - \tilde{E}_1)\tilde{F}_1 - (\tilde{F}_n - \tilde{F}_1)\tilde{E}_1$;
for each point $i$ do
    $d_i \leftarrow \frac{a\tilde{E}_i + b\tilde{F}_i + c}{\sqrt{a^2+b^2}}$ ;                // Distance to baseline
end
$i^*_{\text{Kneedle}} \leftarrow \arg\max_i d_i$;
$w^*_{\text{Kneedle}} \leftarrow w_{i^*}$;
// Method 2: Marginal Cost Analysis;
for $i = 2$ to $n$ do
    $\Delta F \leftarrow F_i - F_{i-1}$;
    $\Delta E \leftarrow E_i - E_{i-1}$;
    $\text{MC}_i \leftarrow \begin{cases} -\Delta F/\Delta E & |\Delta E| > 0.001 \\ 0 & \text{otherwise} \end{cases}$;
end
$\text{MC}_{\max} \leftarrow \max_i \text{MC}_i$;
$i^*_{\text{marginal}} \leftarrow \min\{i : \text{MC}_i < 0.5 \cdot \text{MC}_{\max} \wedge \text{MC}_{i-1} \geq 0.5 \cdot \text{MC}_{\max}\}$;
$w^*_{\text{marginal}} \leftarrow w_{i^*}$;
return $w^*_{\text{Kneedle}}$, $w^*_{\text{marginal}}$;
```

Algorithm 21: Dynamic Sigmoid Weight Function with Grid Search

```
Input: Knee points $w^*_{\min} \approx 0.2$, $w^*_{\max} \approx 0.6$, parameter grids
Output: Optimal sigmoid parameters $(w_{\min}, w_{\max}, k)$
// Sigmoid Weight Function;
Function SigmoidWeight($t$, $t_{\max}$, $w_{\min}$, $w_{\max}$, $k$):
    $t_{\text{mid}} \leftarrow t_{\max}/2$;
    $t_{\text{norm}} \leftarrow (t - t_{\text{mid}})/(t_{\max}/4)$;
    return $w_{\min} + \frac{w_{\max} - w_{\min}}{1+\exp(-k \cdot t_{\text{norm}})}$;
// Grid Search Optimization;
$\mathcal{G} \leftarrow \{0.2, 0.3, 0.4\} \times \{0.6, 0.7, 0.8\} \times \{0.5, 1.0, 1.5, 1.9, 2.0\}$;
for each $(w_{\min}, w_{\max}, k) \in \mathcal{G}$ with $w_{\min} < w_{\max}$ do
    for each season $s$ do
        for each week $t$ do
            $w(t) \leftarrow$ SigmoidWeight($t$, $t^s_{\max}$, $w_{\min}$, $w_{\max}$, $k$);
            Simulate elimination with weight $w(t)$;
        end
    end
    Compute $(F, E, R)$ for this parameter set;
    composite $\leftarrow 0.4 \cdot F + 0.3 \cdot E + 0.3 \cdot R$;
end
$(w^*_{\min}, w^*_{\max}, k^*) \leftarrow \arg\max$ composite;
return Optimal parameters: $w^*_{\min} = 0.2$, $w^*_{\max} = 0.6$, $k^* = 1.9$;
```

Algorithm 22: Hybrid Dynamic Weight with Week-7 Bottom-2 Rule

```
Input: Season data, sigmoid params $(w_{\min}, w_{\max}, k)$, switch week
        $t_{\mathrm{B2}} = 7$
Output: Elimination decision for each week
for each week $t$ in season do
    $w(t) \leftarrow$ SigmoidWeight$(t, t_{\max}, w_{\min}, w_{\max}, k)$;
    $T_i \leftarrow w(t) \cdot \tilde{s}_i + (1 - w(t)) \cdot \tilde{f}_i$ for all $i$;
    if $t < t_{B2}$ then
        // Standard elimination:  lowest total score
        $e \leftarrow \arg\min_i T_i$;
    else
        // Bottom-2 rule:  judges decide between lowest two
        $(b_1, b_2) \leftarrow$ two contestants with lowest $T_i$;
        if $s_{b_1} < s_{b_2}$ then
            $e \leftarrow b_1$ ;                // Judges eliminate lower scorer
        else
            $e \leftarrow b_2$;
        end
    end
    Record elimination $e$ and metrics;
end
return Season elimination sequence, fairness/engagement metrics;
```

Algorithm 23: Controversial Case Validation with Proposed System

```
Input: Historical cases $\mathcal{C} = \{$Bobby Bones, Jerry Rice, Sabrina Bryan,
       Bristol Palin$\}$
Output: Comparison of outcomes under current vs proposed system
// Define Systems;
$\mathcal{S}_{\text{current}} \leftarrow w = 0.5$ (fixed);
$\mathcal{S}_{\text{proposed}} \leftarrow w(t) = 0.2 + 0.4/(1 + \exp(-1.9 \cdot t_{\text{norm}}))$;
// Simulate Each Case;
for each case $(c, s) \in \mathcal{C}$ do
    $\mathcal{D}_s \leftarrow$ season $s$ data;
    for each week $t$ contestant $c$ participated do
        // Current system simulation
        Rank $c$ by combined score with $w = 0.5$;
        $r_{\text{current}}(t) \leftarrow$ combined rank of $c$;
        // Proposed system simulation
        $w(t) \leftarrow \texttt{SigmoidWeight}(t, t_{\max}, 0.2, 0.6, 1.9)$;
        Rank $c$ by combined score with $w(t)$;
        $r_{\text{proposed}}(t) \leftarrow$ combined rank of $c$;
        $\text{at\_risk}_{\text{current}}(t) \leftarrow \mathbf{1}[r_{\text{current}}(t) = n]$;
        $\text{at\_risk}_{\text{proposed}}(t) \leftarrow \mathbf{1}[r_{\text{proposed}}(t) = n]$;
    end
    Store trajectory $\{r_{\text{current}}(t), r_{\text{proposed}}(t)\}_{t=1}^{T_c}$;
end
// Count Robberies System-Wide;
for each system $\mathcal{S} \in \{\mathcal{S}_{current}, \mathcal{S}_{proposed}\}$ do
    $N^{\mathcal{S}}_{\text{robbery}} \leftarrow |\{(s, w) :$ eliminated had judge rank $\leq 2\}|$;
end
$\Delta_{\text{robbery}} \leftarrow N^{\text{current}}_{\text{robbery}} - N^{\text{proposed}}_{\text{robbery}}$;
return Case trajectories, robbery reduction $\Delta_{\text{robbery}}$;
```

Algorithm 24: Composite Benefit Function Optimization

```
**Input:** Objective values $(F, E, R)$ for each system configuration
**Output:** Optimal configuration maximizing total benefit
// **Define Composite Benefit**;
$B(F, E, R) = \alpha_F \cdot F + \alpha_E \cdot E + \alpha_R \cdot R$;
where $\alpha_F = 0.4$, $\alpha_E = 0.3$, $\alpha_R = 0.3$;
// **Compare Configurations**;
$B_{\text{current}} \leftarrow B(F_{w=0.5}, E_{w=0.5}, R_{w=0.5})$;
$B_{\text{sigmoid}} \leftarrow B(F_{\text{sigmoid}}, E_{\text{sigmoid}}, R_{\text{sigmoid}})$;
$B_{\text{hybrid}} \leftarrow B(F_{\text{hybrid}}, E_{\text{hybrid}}, R_{\text{hybrid}})$;
// **Select Best System**;
$\mathcal{S}^* \leftarrow \arg\max_{\mathcal{S}} B(\mathcal{S})$;
// **Quantify Improvements**;
$\Delta_{\text{fairness}} \leftarrow F^* - F_{\text{current}}$;
$\Delta_{\text{engagement}} \leftarrow E^* - E_{\text{current}}$;
$\Delta_{\text{robbery}} \leftarrow R^* - R_{\text{current}}$;
**return** Optimal system $\mathcal{S}^*$, improvements $(\Delta_F, \Delta_E, \Delta_R)$;
```

Algorithm 25: Temporal Sensitivity Analysis: Era Comparison

```
Input: Feature data $\mathcal{D}$, era split point $s_{\text{split}} = 15$
Output: Era-specific coefficients and stability assessment
// Split Data by Era;
$\mathcal{D}_{\text{early}} \leftarrow \{d \in \mathcal{D} : s_d \leq s_{\text{split}}\}$ ;                // S1-S15
$\mathcal{D}_{\text{late}} \leftarrow \{d \in \mathcal{D} : s_d > s_{\text{split}}\}$ ;                 // S16-S31
// Define LMM Formula;
Formula: score_zscore $\sim$ age + $\mathbf{C}$(Industry) + week;
Random effect: (ballroom_partner);
// Fit LMM for Each Era;
for each dataset $\mathcal{D}_e \in \{\mathcal{D}_{full}, \mathcal{D}_{early}, \mathcal{D}_{late}\}$ do
    Fit mixed model: score = $\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\epsilon}$;
    // Extract key coefficients
    $\beta_{\text{age}}^{(e)} \leftarrow$ coefficient for age;
    $\text{CI}_{\text{age}}^{(e)} \leftarrow$ 95% confidence interval;
    $\beta_{\text{Model}}^{(e)} \leftarrow$ coefficient for Industry = Model;
    $\text{CI}_{\text{Model}}^{(e)} \leftarrow$ 95% confidence interval;
    // Calculate Intraclass Correlation
    $\sigma^2_{\text{partner}} \leftarrow \text{Var}(\mathbf{u})$;
    $\sigma^2_{\epsilon} \leftarrow \text{Var}(\boldsymbol{\epsilon})$;
    $\text{ICC}^{(e)} \leftarrow \sigma^2_{\text{partner}}/(\sigma^2_{\text{partner}} + \sigma^2_{\epsilon})$;
    Store $(\beta_{\text{age}}^{(e)}, \beta_{\text{Model}}^{(e)}, \text{ICC}^{(e)}, \text{CI})$;
end
// Assess Stability;
$\Delta_{\text{age}} \leftarrow \beta_{\text{age}}^{\text{late}} - \beta_{\text{age}}^{\text{early}}$;
$\Delta_{\text{Model}} \leftarrow \beta_{\text{Model}}^{\text{late}} - \beta_{\text{Model}}^{\text{early}}$;
$\Delta_{\text{ICC}} \leftarrow \text{ICC}^{\text{late}} - \text{ICC}^{\text{early}}$;
// Robustness criteria
Stable $\leftarrow (\text{sign}(\beta_{\text{age}}^{\text{early}}) = \text{sign}(\beta_{\text{age}}^{\text{late}})) \wedge (\text{sign}(\beta_{\text{Model}}^{\text{early}}) = \text{sign}(\beta_{\text{Model}}^{\text{late}}))$;
return Era coefficients, confidence intervals, $(\Delta_{\text{age}}, \Delta_{\text{Model}}, \Delta_{\text{ICC}})$,
  Stable;
```

Algorithm 26: Parameter Sensitivity Analysis

```
**Input:** Data $\mathcal{D}$, $w_{\max} \in [0.5, 0.7]$, $k \in [1.0, 3.0]$
**Output:** Sensitivity heatmap $\mathbf{M}$, optimal parameters
Initialize grid $\mathbf{M}_{21\times 21}$, $w_{\min} \leftarrow 0.20$;
for each $w_{\max} \in linspace(0.50, 0.70, 21)$ do
    for each $k \in linspace(1.0, 3.0, 21)$ do
        for each $(s, t) \in \mathcal{D}$ do
            $w(t) \leftarrow w_{\min} + (w_{\max} - w_{\min})/(1 + \exp(-k \cdot t_{\text{norm}}))$;
            $T_i \leftarrow w(t) \cdot \tilde{s}_i + (1 - w(t)) \cdot \tilde{f}_i$;
        end
        $F \leftarrow$ mean relative rank of eliminated;
        $R \leftarrow$ robbery rate (judge rank $\leq 2$);
        $E \leftarrow 1 - \bar{w}$;
        $\mathbf{M}[w_{\max}, k] \leftarrow 0.4F + 0.3E + 0.3(1 - R)$;
    end
end
$(w^*_{\max}, k^*) \leftarrow (0.60, 1.90)$ from optimal region;
return Heatmap $\mathbf{M}$, optimal $(w^*_{\max}, k^*)$;
```

Algorithm 27: Monte Carlo Noise Robustness Analysis

```
**Input:** Season data, $N_{\text{sim}} = 50000$, $\sigma_{\text{noise}} = 0.15$, targets $\mathcal{T}$
**Output:** Rank trajectories with 95% CI, danger probabilities
for each target $(c, s) \in \mathcal{T}$ do
    $\mathcal{D}_s \leftarrow$ season $s$ data, initialize $\mathbf{R}_c$;
    for $i = 1$ to $N_{sim}$ do
        $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2_{\text{noise}})$;
        $\tilde{f}_{\text{noisy}} \leftarrow \text{clip}(\tilde{f} \cdot (1 + \boldsymbol{\epsilon}), 0.001, 0.999)$;
        $\tilde{f}_{\text{norm}} \leftarrow \tilde{f}_{\text{noisy}} / \sum_j \tilde{f}_{\text{noisy},j}$;
        for each week $t$ do
            $w(t) \leftarrow \texttt{SigmoidWeight}(t, t_{\max}, 0.20, 0.60, 1.90)$;
            $T_i \leftarrow w(t) \cdot \tilde{s}_i + (1 - w(t)) \cdot \tilde{f}_{\text{norm},i}$;
            $r^{(i)}_c(t) \leftarrow$ rank of $c$;
        end
        Append $\{r^{(i)}_c(t)\}_t$ to $\mathbf{R}_c$;
    end
    for each week $t$ do
        $\bar{r}_c(t) \leftarrow \text{mean}(\mathbf{R}_c[:, t])$;
        $\text{CI}_{95}(t) \leftarrow [\text{quantile}_{0.025}, \text{quantile}_{0.975}]$;
        $P_{\text{danger}}(t) \leftarrow \Pr(r_c(t) \leq 2)$;
    end
end
return Trajectories $\{\bar{r}_c(t), \text{CI}_{95}(t)\}$, $P_{\text{danger}}$;
```