Mapping the Latent Structure of Economic Bias: A DML-Unfolding Fusion Framework

Model-Heavy

Tyler Venner*, Macy Chen, Leonel Garibay-Estrada, Jiarui Hou, Alberto Ramirez[1]

[1]Team 5, University of California, Davis

November 28, 2025

Abstract

This document details a novel, two-stage methodological framework for visualizing the high-dimensional economic bias in professional sports markets. Traditional econometric analyses often conclude with a static list of significant coefficients that fails to reveal the latent relationships between bias factors or the systemic clusters of affected individuals. Furthermore, standard models often ignore the confounding influence of deterministic salary structures, such as Rookie Scales and Maximum contracts, leading to model misspecification. To address these limitations, we introduce a Stratified Double Machine Learning (DML) approach. We first isolate the Free Market sub-population to learn the performance-debiased "price" of contextual factors (Z). These coefficients are then counterfactually applied to the full population to construct a Debiased Attribution Matrix (L). The probabilistic dimension reduction model of this matrix reveals the geometric structure of bias, clustering factors based on shared structural forces and identifying player archetypes most vulnerable to market inefficiencies.

Keywords: Double Machine Learning, Latent Space Mapping, Sports Analytics, Economic Bias, Dimensionality Reduction

1 Introduction & Motivation

Standard econometric analysis for quantifying discrimination and market inefficiency, while statistically correct, often suffers from an interpretive bottleneck. These methods typically conclude with a "wall of coefficients", a static table of estimates and p-values that answers the binary question of statistical significance. This fails to reveal the underlying system of bias. For example, while a regression may indicate that both Age and Draft Number significantly impact salary, it cannot intuitively demonstrate whether these factors act as independent forces or reveal how these biases affect individual players relative to one another.

Furthermore, applying standard statistical models to professional sports markets presents a unique structural challenge: the prevalence of deterministic contracts. In the National Basketball Association (NBA), a substantial subset of player salaries, specifically Rookie Scale and Maximum contracts, are dictated by Collective Bargaining Agreement (CBA) rules rather than free-market negotiation. Applying a standard bias model to this mixed population will result in severe model misspecification. For instance, a high-performing rookie may appear to have a large negative residual not because of market bias, but due to a salary ceiling.

To address these interpretive and structural limitations, this report introduces a novel DML-Dimension Reduction Framework. We use Double Machine Learning (DML) to strictly isolate the price of bias factors, and our custom dimension reduction technique to visualize their geometric relationships.

Our approach moves the goal from simply quantifying bias to mapping its topology. By implementing a stratified Learn vs. Apply protocol, we first learn the true market valuation from free-market players and then counterfactually apply these valuations to the entire league. The resulting Bias Attribution Map reveals the latent structure of economic forces, identifying distinct player clusters and the specific structural biases that drive their valuation.

2 Data and Methods

Our methodological framework fuses inference with geometric visualization. We employ a two-stage approach:

first, we isolate the "price" of bias using Stratified Double Machine Learning (DML), and second, we map the topology of these biases using our dimension reduction method.

2.1 Data

Our dataset integrates player performance, contract valuation, team ownership, market characteristics and demographic attributes to compile a comprehensive overview of factors that contribute to salary determination for the 2024-2025 season.

Player-level performance metrics (X) were obtained from NBA API and include advanced offensive and defensive indicators such as offensive rating, defensive rating, net rating, assist percentage, rebound percentage, etc. Salary and contract information (Y) were scraped from Spotrac.com, capturing each player's total annual compensation. To model contextual bias factors (Z), we collected draft position, draft round, age, and nationality, as well as team-level attributes such as team salary cap utilization, ownership net worth, and stadium capacity. Player social media presence was captured using Instagram followers count collected from popularbasketballers.com.

The final dataset contains one row per player, representing each player's aggregated performance, demographic and contractual profile.

2.2 The Learn-Apply Protocol

A critical challenge in analyzing NBA salaries is the prevalence of non-market compensation. Training a model on the full population (D_{ALL}) would allow deterministic salaries (Rookie Scale and Maximum contracts) to contaminate the estimate of market valuation.

To resolve this, we implement a stratified Learn-Apply protocol:

1. Learn Phase: We isolate the sub-population of players operating under negotiated, free-market contracts (D_{FM}). We train the confounding models (specifically, the mappings from performance to salary and bias factors) solely on this group to capture true market dynamics.

2. Apply Phase: We apply these learned baseline models and the resulting market prices to the entire population (D_{ALL}), creating counterfactual estimates for players on fixed contracts.

2.3 Stage 1: Stratified Double Machine Learning

We assume a partially linear structural model for player valuation:

$$Y_i = \alpha + Z_i^T \gamma + f(X_i) + \epsilon_i \tag{1}$$

where Y is log-salary, Z is the vector of bias factors, X is the vector of performance metrics, and γ represents the marginal market price of bias.

To estimate γ unbiasedly, we use the machine-learning extension of the Frisch-Waugh-Lovell (FWL) theorem. The central theme of FWL is that to isolate the linear effect of Z on Y in the presence of a confounder X, one must remove the influence of X from *both* the outcome and the treatment variables. This process ensures that the final estimate $\hat{\gamma}$ captures only the relationship between the unexplained components of salary (Y) and bias factors (Z), orthogonal to performance (X).

Crucially, to prevent overfitting bias, we do not generate residuals on the same data used to train the models. Instead, we employ K-fold Cross-Fitting (Sample Splitting). We randomly partition the free-market data D_{FM} into $K = 5$ folds. For each fold k, we train the models on the complementary folds ($K \setminus k$) and generate out-of-sample residuals for the players in fold k. These residuals are then stacked to form the final vectors used in the regression stage.

For each player $i \in D_{FM}$, the procedure involves three steps (performed via cross-fitting):

1. Outcome Model (Cleaning Y): We train a gradient boosting model $\hat{f}(X)$ to predict salary. This ensures the residual $\hat{\epsilon}_{Y,i}$ represents salary strictly unexplained by performance.

$$\hat{\epsilon}_{Y,i} = Y_i - \hat{f}(X_i) \tag{2}$$

2. Treatment Models (Cleaning Z): For each bias factor Z_j, we train a separate boosting model $\hat{h}_j(X)$ to predict the factor from performance. The residual represents the "pure" component of the factor (e.g., the part of Draft Number (Z_j) not predicted by performance (X)):

$$\hat{\epsilon}_{Z^{(j)},i} = Z_i^{(j)} - \hat{h}_j(X_i) \tag{3}$$

3. Debiased Regression: Finally, we estimate the bias coefficients by regressing the *out-of-sample* outcome residuals onto the treatment residuals via simple OLS:

$$\hat{\epsilon}_{Y,i} = \gamma_0 + \sum_{j=1}^{m} \gamma_j \hat{\epsilon}_{Z_j,i} + \nu_i \tag{4}$$

We employ OLS in this final stage rather than a flexible model for theoretical sufficiency and interpretability. Since the complex non-linear confounding has

already been removed by the ML models in steps 1 and 2, the remaining relationship between residuals is structural.

To ensure the confounding models $\hat{f}(X)$ and $\hat{h}_j(X)$ generalize well to unseen data, we employed a tuning protocol. We utilized Scikit-Learn's `GradientBoostingRegressor` within a pipeline that first standardized all inputs via `StandardScaler`.

Optimal hyperparameters were selected using `RandomizedSearchCV` with 5-fold internal cross-validation. Rather than searching an exhaustive grid, we sampled 20 candidate configurations from a parameter space. The final model for each fold was the candidate that maximized the internal cross-validation R^2 score.

2.4 Stage 2: The Attribution Matrix

To connect the output of our ML models with the geometric visualization, we construct the Attribution Matrix $L \in \mathbb{R}^{n \times m}$. Raw bias factors ($Z$) have non comparable units (e.g., years of Age vs. # followers on Instagram), making direct geometric embedding impossible.

First, for every player $k \in D_{ALL}$, regardless of contract status, we calculate the counterfactual residual $\hat{\epsilon}_{Z^{(j)},k}$ using the treatment models trained exclusively on free-market players. For rookies and max-contract players, this quantity represents the "excess bias" they possesses relative to their performance, had they been subject to open negotiation.

We then define each entry L_{ij} as the magnitude of economic impact bias factor j has on player i:

$$L_{ij} = |\hat{\gamma}_j \cdot \hat{\epsilon}_{Z^{(j)},i}| \tag{5}$$

This transformation achieves two objectives:

1. Unit Unification: By interacting the market price ($\hat{\gamma}_j$) with the residual volume ($\hat{\epsilon}_{Z,ij}$), we convert all factors into a single, unified currency: log-dollars of unexplained salary attribution.

2. Relevance over Direction: We utilize the absolute value to shift the analytical focus from direction (overpaid vs. underpaid) to magnitude (relevance). Geometrically, we seek to cluster players whose valuations are driven by the same structural forces, regardless of whether that force acts as a premium or a penalty.

Finally, we apply a global min-max scaling to map these attributions onto the $[0, 1]$ interval required for our probabilistic unfolding algorithm.

2.5 Stage 3: Latent Unfolding (Dimension Reduction)

We visualize the high-dimensional Attribution Matrix L by mapping it to a low-dimensional space, $\mathbb{R}^3$. Unlike traditional multidimensional scaling which treats entities as fixed points, we employ a probabilistic approach which has some nice mathematical properties we will discuss later.

We represent the position of player i as a random variable $p_i \sim \mathcal{N}(\mu_{p,i}, V_{p,i})$ and bias factor j as $b_j \sim \mathcal{N}(\mu_{b,j}, V_{b,j})$. The interaction between a player and a factor is defined by their difference vector $\delta_{ij} = p_i - b_j$, which is itself Gaussian distributed:

$$\delta_{ij} \sim \mathcal{N}(\mu_{ij}, V_{ij}) \tag{6}$$

where $\mu_{ij} = \mu_{p,i} - \mu_{b,j}$ and $V_{ij} = V_{p,i} + V_{b,j}$.

Formulating the locations players and variables on the latent space as probability distributions offers a mathematical advantage over point-based embeddings. If the model is uncertain about a player's location (represented by a large covariance Σ), the density spreads out, lowering the peak similarity score. This forces the model to only cluster entities tightly when the statistical signal is strong enough to justify low variance.

We define the theoretical attribution strength using a Gaussian kernel, which maps geometric proximity to a similarity score bounded in $(0, 1]$:

$$g(\delta) = \exp(-\delta^\top \delta) = \exp(-\|\delta\|_2^2) \tag{7}$$

The model prediction $\hat{L}_{ij}$ is the expected value of the similarity function over the entire distribution of possible relative positions. This is defined by the integral:

$$\hat{L}_{ij} = \mathbb{E}[g(\delta)] = \int_{\mathbb{R}^3} g(\delta) \cdot p(\delta|\mu_{ij}, V_{ij}) d\delta \tag{8}$$

While numerically integrating this over high dimensions is computationally intractable, we notice the fact that this integral corresponds to the moment-generating function (MGF) of a quadratic form of a Gaussian variable. Which has a closed-form solution:

$$\hat{L}_{ij} = |I + 2V_{ij}|^{-1/2} \exp\left(-\mu_{ij}^\top (I + 2V_{ij})^{-1} \mu_{ij}\right) \tag{9}$$

We notice that it naturally penalizes uncertainty: as the variance V_{ij} increases, the expected attribution $\hat{L}_{ij}$ decreases, reflecting lower confidence for that relationship.

We solve for the optimal map coordinates $\theta = \{\mu, \Sigma\}$ using Maximum Likelihood Estimation (MLE). We assume that the observed entries in the attribution matrix L_{ij} are drawn from a Gaussian distribution centered at the model's prediction $\hat{L}_{ij}(\theta)$, with a global noise variance σ_{err}^2:

$$p(L_{ij}|\theta, \sigma_{err}) = \frac{1}{\sqrt{2\pi\sigma_{err}^2}} \exp\left(-\frac{(L_{ij} - \hat{L}_{ij}(\theta))^2}{2\sigma_{err}^2}\right) \tag{10}$$

To find the optimal parameters, we maximize the joint likelihood over all player-factor pairs: $\mathcal{L}(\theta) = \prod_{i,j} p(L_{ij}|\theta)$.

For numerical stability, we instead minimize the Negative Log-Likelihood (NLL). Taking the negative natural logarithm turns the product into a sum and the exponent into a linear term:

$$\begin{aligned}\mathcal{J}(\theta) &= -\log \mathcal{L}(\theta) \\ &= \sum_{i,j} \left(\frac{(L_{ij} - \hat{L}_{ij}(\theta))^2}{2\sigma_{err}^2} + \frac{1}{2}\log(2\pi\sigma_{err}^2) \right) \end{aligned} \quad (11)$$

Dropping constant terms ($\log(2\pi)$), this simplifies to the final objective function:

$$\mathcal{J}(\theta) = \sum_{i,j} \left(\frac{(L_{ij} - \hat{L}_{ij}(\theta))^2}{2\sigma_{err}^2} + \log(\sigma_{err}) \right) \quad (12)$$

We notice that minimizing the NLL is mathematically equivalent to minimizing a weighted Sum of Squared Errors (SSE).

Minimizing this non-convex objective is computationally intensive due to the requirement of calculating determinants and inverses for covariance matrices at every step. Initial implementations using standard sequential processing failed to converge to a stable global minimum and suffered from prohibitive runtime (> 60 seconds per initialization).

To overcome this, we re-implemented the optimization loop using Google's JAX library. This offered three advantages:

1. Automatic Differentiation: JAX computes exact gradients for the complex covariance parameters via `grad`.
2. Vectorization: We utilized `vmap` to compute the moment-generating functions for all $N \times M$ pairs in parallel, eliminating Python loops which are slow.
3. Just-In-Time (JIT) Compilation: The step function was compiled into machine code, which runs much faster than python code.

This architecture reduced the optimization time from over one minute to less than one second and it ensures convergence.

3 Results & Interpretation

3.1 Stage 1 Diagnostics: Validating the "Fair" Price

Before analyzing bias, we validated that our Stage 1 Gradient Boosting model ($\hat{f}(X)$) correctly captured the "fair" market value of performance. The model achieved an R^2 of 0.483, indicating that approximately 48% of salary variance is explained purely by on-court statistics.

The feature importance analysis (Table 1) shows the importance of top features. We calculated these scores using the Gini Index. This metric quantifies the total reduction in the variance (squared error) brought about by a feature across all decision trees in the ensemble. A higher score implies the feature was frequently used to make significant distinctions in player valuation.

The analysis reveals the top predictors were "Clutch Points" (performance in close games), "Net Rating" (overall team impact), and "Minutes Played" (availability), whereas volume stats like "Field Goals Attempted" were less dominant.

Table 1: Top 5 Performance Drivers of Salary ($\hat{f}(X)$)

Feature	**Importance Score**
Clutch Points	0.184
Net Rating	0.091
Minutes	0.090
Clutch GP	0.080
FGM per Game	0.075

3.2 Econometric Estimation: The "Wall of Coefficients"

The DML procedure yielded the debiased market prices ($\hat{\gamma}$) for the contextual factors. As detailed in Table 2, we identified specific statistically significant drivers of salary deviation.

Table 2: Learned Market Prices of Bias (Stage 1 Output)

Factor	**Price ($\hat{\gamma}$)**	**SE**	**p-val**
Age	**0.0801**	**0.021**	**<0.001**
Followers	**0.0432**	**0.019**	**0.022**
Active Cap	-0.0000	0.000	0.049
is_USA	-0.2088	0.126	0.098
Stadium Year	-0.0132	0.009	0.161
Owner Net Worth	-0.0667	0.068	0.324
Draft Number	0.0016	0.004	0.695
Stadium Cost	0.0000	0.000	0.994

The top two drivers of market inefficiency:

1. The Veteran Premium: 'Age' is the most significant factor ($p < 0.001$). The coefficient $\hat{\gamma}_{age} \approx 0.08$ indicates that for every additional year of age, a player's salary increases by approximately **8.0%**, holding strictly constant for on-court performance. Notably, the nuisance model for Age achieved an $R^2 \approx 0$, confirming that Age is orthogonal to performance statistics. This suggests veteran status provides a premium regardless of declining physical output.

2. The Fame Premium: 'Followers' is statistically significant ($p = 0.022$). The positive coefficient suggests a bias where a one-unit increase in social media following is associated with a 4.3% increase in valuation, independent of the player's actual contribution to winning.

Additionally, team-level financial constraints played a role, with 'Active Cap' showing borderline significance ($p = 0.049$).

Crucially, Draft Number was statistically insignificant ($p = 0.695$). This implies that in the Free Market (D_{FM}), teams effectively "forget" draft pedigree. Once a player reaches their second contract, the "lottery pick" prestige evaporates, and valuation is driven by realized performance, age, and fame.

3.3 Topological Structure: The Bias Map

The application of the probabilistic unfolding algorithm to the Attribution Matrix L transformed these static coefficients into a geometric map (for interactive visualization, see the web page). This visualization reveals a topology defined by significance and contract status.

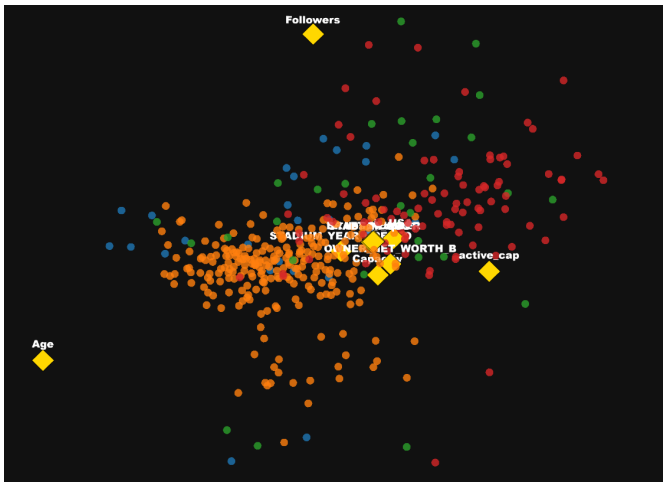

Figure 1: The 3D Bias Attribution Map. Yellow diamonds represent bias factors; dots represent players colored by contract type. Note the clustering of Free Market players (Orange) near Age, and Rookies (Red) separated from it. Explore the map on the website. See bigger image in appendix.

1. Significance Sorting. The map automatically sorted factors by their economic relevance. Factors with statistically insignificant coefficients, specifically `DRAFT_NUMBER`, `is_USA`, and `OWNER_NET_WORTH`, collapsed into a tight cluster in the geometric center of the map. Because their market price γ is approximately zero, they exert near-zero gravitational pull, effectively creating a "null space" for bias. In contrast, the statistically significant factors, `Age` (positioned to the left), `active_cap` (top), and `Followers` (bottom), were pushed to the periphery, acting as the distinct "poles" of the market.

2. The Age Axis and Contract Separation. Overlaying contract types revealed a stark structural separation along the horizontal axis, driven almost entirely by the `Age` anchor. Players on negotiated Free Market contracts (orange) clustered heavily on the left side of the map, appearing as the primary "consumers" of the Age Premium. This confirms that once players leave the rookie scale, their valuation becomes structurally tied to their tenure. Conversely, players on the Rookie Scale (red) formed a distinct cluster on the right, separated from the Age anchor.

3. Archetypal Validation. Finally, the interaction between the `Age` and `Followers` anchors creates a specific region in the bottom-left quadrant. This area represents the "Aging Superstar" archetype, players who command premiums for both tenure and social capital (e.g., LeBron James), independent of their current on-court production.

4 Discussion and Reflection

The development of the DML-Unfolding framework presented significant methodological challenges, primarily centered on the incompatibility of standard econometric outputs with geometric visualization tools.

4.1 Bridging the Analytical Gap

The primary innovation of this work lies in the fact that traditional DML workflows terminate at the estimation of $\hat{\gamma}$, effectively treating the bias coefficients as the final answer. However, we identified that these coefficients are merely the "price" in a larger system. The challenge was to convert these scalar prices into a format suitable for the dimension reduction algorithm.

Our solution, the Debiased Attribution Matrix ($L_{ij} = |\hat{\gamma}_j \cdot \hat{\epsilon}_{Z,ij}|$), proved critical. By interacting the price of bias with the magnitude of the residual, we successfully unified disparate units (e.g., years of age vs. number of followers) into a single, comparable currency of "unexplained log-dollars". Without this normalization, the latent space map would have been distorted by the arbitrary scaling of the input features rather than their true economic influence.

4.2 Limitations and Transportability

A central limitation of our stratified approach is the reliance on the "Transportability Assumption": we assume that the price of bias learned from the free-market subpopulation (D_{FM}) is structurally stable and applies counterfactually to players on deterministic contracts (Rookies, Max, Minimum).

While restricting the analysis exclusively to free-market players would have eliminated this extrapolation risk, it would have rendered the framework incapable of diagnosing league-wide structural dynamics. Conversely, including deterministic players in the training set constitutes severe model misspecification. For a rookie on a fixed scale or a star on a Max contract, salary is a function of CBA rules, not market valuation. A standard regression would misinterpret a salary cap as a "bias against stars" or a rookie scale as a "market penalty," confounding institutional constraints with genuine economic preference.

Our stratified Learn-Apply protocol is the theoretically correct solution to this dilemma. By learning prices solely from the free market, we isolate the pure economic signal. By applying them counterfactually, we reveal the "shadow price" of bias—demonstrating, for example, how much a rookie *would* be devalued for their age if they were not protected by the salary scale. This transforms the map from a simple descriptive plot into a diagnostic tool that disentangles market signal from institutional noise.

Nevertheless, this comes with the risk of extrapolation. If the performance profiles of rookies differ systematically from veterans (e.g., lower efficiency or minutes), the treatment models $\hat{h}_{FM}$ may struggle to accurately predict their "fair" baseline. Future iterations should incorporate covariate overlap tests to quantify the statistical validity of these specific counterfactuals.

4.3 Practical Implications

Despite these limitations, the framework offers immediate diagnostic value. Stakeholders can now distinguish between "Pure Performance" players, who reside in the map's center and are valued efficiently, and those subject to heavy structural distortion. For General Managers, this provides a tool to identify market inefficiencies not just by magnitude, but by type, revealing whether a player's valuation is driven by sustainable on-court production or transient structural factors like "Hype" or "Market Size".

4.4 Challenges and Fallback Strategies

Listed below is a few of the technical challenges and their solutions we faced throughout project.

- The most significant hurdle was the optimization of the probabilistic map. Our initial implementation using standard Python loops and SciPy solvers failed to converge to a stable global minimum and suffered from prohibitive runtimes (> 60 seconds). To solve this, we fully migrated to Google's JAX library and using their Just-in-Time (JIT) Compilation and automatic differentiation. This reduced optimization time to below a second and ensured convergence.
- A major theoretical hurdle was trying to find and develop a rigorous statistical method to visualize the bias map. We did not want to stop at "a wall of coefficients", as stated early. An insight came when we understood the distance between the bias variables themselves with the individual players could connect to how the bias explains that particular player. This was the conceptual leap that lead to our dimension reduction solution and a connection from geometry to bias. We then solved the different units problem with our bias attribution matrix ($L_{ij} = |\hat{\gamma}_j \cdot \hat{\epsilon}_{ij}|$).
- Merging data collected from many disparate sources (NBA API, Spotrac, Instagram) was challenging due to inconsistent formats, missing values, and conflicting identifiers. As a solution, we modularized our data pipeline, standardizing and validating each dataset independently before integration, which made debugging easier. In some cases we had to manually go into the data and variable formats to ensure that merging went smoothly.

4.5 Future Work

The "Bias Attribution Map" serves as a powerful new analytical object, opening several avenues for future research. First, a longitudinal extension of this framework could estimate bias coefficients γ_t for multiple seasons, revealing whether the geometric structure of bias is stable or evolves in response to rule changes. Second, this DML-Unfolding fusion is domain-agnostic; it can be applied to other high-stakes fields where "treatment effects" can be attributed, such as mapping property feature biases in real estate or candidate attribute biases in algorithmic hiring.

5 Acknowledgment and References

In developing our methodological framework, Large Language Models (LLMs), including ChatGPT, helped us clarify ideas, debug code, and improve the readability of our explanations. These tools did not generate data, run analyses, or design the core methods. Instead, they served as helpful references that supported our understanding and communication of the material. The sources we used to gather data were NBA API, spotrac.com, popularbasketballers.com, (cont.).

We also acknowledge the open-source community for the Python libraries used in this analysis, specifically Google's `JAX` for the optimization pipeline and `Scikit-Learn` for the machine learning implementation.

Appendix

A Supplementary Visuals

This appendix provides a high-resolution, expanded view of the latent structure visualizations referenced in Section 3.3.

Figure 2 (below) offers a detailed view of the "Bias Attribution Map" introduced in Figure 1. In this enlarged view, the structural separation between the "Free Market" cluster (Orange, left) and "Rookie Scale" cluster (Red, right) is more clearly visible, as is the "Void" of insignificant factors in the center.

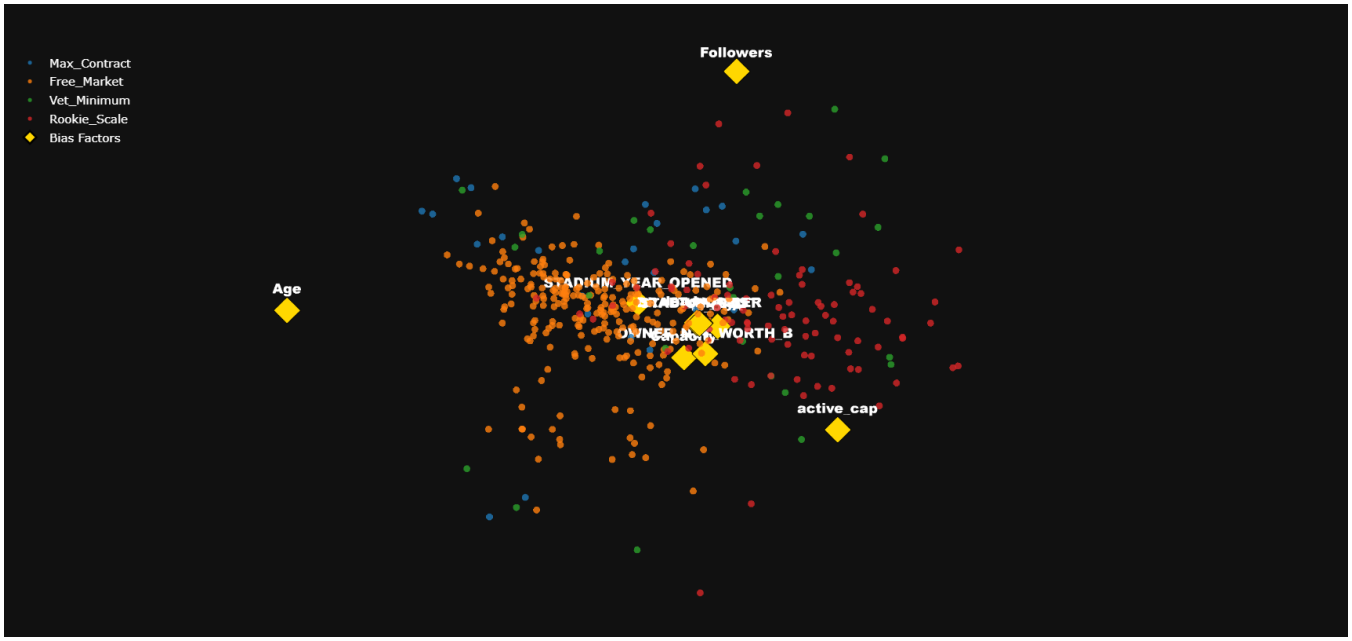

Figure 2: Expanded Bias Attribution Map. This high-resolution view details the "Contract Tectonics" discussed in the Results section. Note the explicit separation of the Rookie cluster (Red) from the Age Anchor (Yellow, Left).

B Implementation Details (Pseudocode)

This section provides the algorithmic specifications for the three core components of the DML-Unfolding framework. You can provide our full code in our public Github Repository.

B.1 Stage 1: DML Pipeline with Hyperparameter Tuning

Technical Summary: Algorithm 1 implements the "Learn" phase of our protocol. The computational complexity is dominated by the training of the Gradient Boosting Machines (GBM). For a dataset of size N with M bias factors and K folds, the complexity is approximately $O(K \cdot (M+1) \cdot T_{train})$, where T_{train} is the cost of fitting a GBM. To ensure the orthogonality condition $E[\epsilon \cdot X] = 0$ required by the Frisch-Waugh-Lovell theorem, we enforce strict sample splitting via K-fold cross-fitting. This prevents "own-observation bias," where the model overfits to the training residuals, which would otherwise bias the final $\hat{\gamma}$ estimates downwards.

Algorithm 1: Stratified Double Machine Learning (DML) with Cross-Fitting

Require: Free Market Data $D_{FM} = \{Y, X, Z\}$, Number of Folds $K = 5$
Require: Hyperparameter Grid $\mathcal{G}$ (Depth, Learning Rate, Subsample)
1: **Initialize:** Empty vectors for residuals $\hat{\epsilon}_Y$, $\hat{\epsilon}_Z$ of length N

Partition D_{FM} into K disjoint folds $\{I_1, \ldots, I_K\}$
for $k = 1$ to K **do**
    Data Splitting:
    $Train_k \leftarrow D_{FM} \setminus I_k$ ▷ Complementary folds
    $Test_k \leftarrow I_k$ ▷ Current fold (Out-of-Sample)
    Step 1: Outcome Model ($\hat{f}$)
    $\hat{f}_k \leftarrow \text{RandomizedSearch}(\text{GBM}, \mathcal{G}, Train_k(X, Y))$ ▷ Tune & Train
    $\hat{Y}_k \leftarrow \text{Predict}(\hat{f}_k, Test_k(X))$
    $\hat{\epsilon}_Y[I_k] \leftarrow Test_k(Y) - \hat{Y}_k$ ▷ Store OOS Residuals
    Step 2: Treatment Models ($\hat{h}$)
    for each bias factor $j \in \{1, \ldots, M\}$ **do**
        $\hat{h}_{j,k} \leftarrow \text{Train}(\text{GBM}, Train_k(X, Z^{(j)}))$
        $\hat{Z}_k^{(j)} \leftarrow \text{Predict}(\hat{h}_{j,k}, Test_k(X))$
        $\hat{\epsilon}_{Z^{(j)}}[I_k] \leftarrow Test_k(Z^{(j)}) - \hat{Z}_k^{(j)}$
    end for
end for
Step 3: Debiased Regression
$\hat{\gamma} \leftarrow \text{OLS}(\hat{\epsilon}_Y \sim \text{const} + \hat{\epsilon}_Z)$ ▷ Regress stacked residuals
return Bias Coefficients $\hat{\gamma}$, Residuals $\hat{\epsilon}_Z$

B.2 Stage 2: Calculating Debiased Attribution Matrix (L)

Technical Summary: Algorithm 2 acts as the bridge between the econometric estimation (Stage 1) and the geometric visualization (Stage 3). It performs the counterfactual application of the learned market prices $\hat{\gamma}$ to the full population D_{ALL}. Theoretically, this algorithm solves the unit incompatibility problem by projecting disparate feature spaces (e.g., Years, Followers) into a unified currency space (Log-Dollars). The transformation $L_{ij} = |\hat{\gamma}_j \cdot \hat{\epsilon}_{ij}|$ ensures that the resulting geometry is driven by the *magnitude* of economic distortion rather than its direction, allowing for the clustering of players affected by similar structural forces regardless of whether they are over- or under-paid.

Algorithm 2: Construction of the Debiased Attribution Matrix (L)

Require: Full Dataset $D_{ALL} = \{X, Z\}$ (All N players)
Require: Learned Market Prices $\hat{\gamma}$ (Vector of size M)
Require: Trained Treatment Models $\mathcal{H} = \{\hat{h}_1, \ldots, \hat{h}_M\}$ (from Free Market)
Initialize: Empty matrix $\mathcal{E}_Z$ of size $N \times M$
Step 1: Generate Counterfactual Residuals
for each bias factor $j \in \{1, \ldots, M\}$ **do**
    $\hat{Z}^{(j)} \leftarrow \text{Predict}(\hat{h}_j, D_{ALL}[X])$
    if $\hat{h}_j$ is Classifier **then**
        $\hat{Z}^{(j)} \leftarrow \text{PredictProba}(\hat{h}_j, D_{ALL}[X])_{[:,1]}$ ▷ Use probability
    end if
    $\mathcal{E}_Z[:, j] \leftarrow D_{ALL}[Z^{(j)}] - \hat{Z}^{(j)}$ ▷ Residuals for ALL players
end for
Step 2: Calculate Economic Attribution
Initialize: Empty matrix L_{raw} of size $N \times M$
for each player $i \in \{1, \ldots, N\}$ **do**
    for each factor $j \in \{1, \ldots, M\}$ **do**
        $L_{raw}[i, j] \leftarrow |\hat{\gamma}_j \cdot \mathcal{E}_Z[i, j]|$ ▷ Price $\times$ Volume
    end for
end for
Step 3: Global Normalization

$v_{min} \leftarrow \min(L_{raw})$, $v_{max} \leftarrow \max(L_{raw})$
$L \leftarrow \frac{L_{raw} - v_{min}}{v_{max} - v_{min}}$ ▷ Scale to $[0, 1]$ for Map
return Attribution Matrix L

B.3 Stage 3: Latent Unfolding Optimization

Technical Summary: Algorithm 3 minimizes the Negative Log-Likelihood (NLL) of the observed attributions given the latent coordinates. A critical innovation here is the use of the Moment-Generating Function (MGF) to derive a closed-form solution for the expected attribution integral $\mathbb{E}[g(\delta)]$. Without this algebraic shortcut, the objective function would require expensive Monte Carlo integration at every gradient step. We implement this using JAX to leverage Just-In-Time (JIT) compilation, which fuses the linear algebra operations into a single XLA kernel, and automatic differentiation (`grad`), which computes exact derivatives for the covariance matrices Σ without numerical instability.

Algorithm 3: Latent Unfolding (JAX Optimization)

Require: Attribution Matrix $L \in \mathbb{R}^{N \times M}$
Require: Dimensions $d = 3$, Global Noise σ_{err}
Ensure: Latent Coordinates μ_p, μ_b and Variances Σ_p, Σ_b
1. Initialization (Cold Start)
$\mu_p^{(0)}, \mu_b^{(0)} \leftarrow \text{PCA}(L, n\_components = d)$ ▷ Initialize means via PCA
$\Sigma_p^{(0)}, \Sigma_b^{(0)} \leftarrow \text{InitSmall}(1e^{-2})$ ▷ Initialize variances low
$\theta \leftarrow \{\mu_p, \mu_b, \Sigma_p, \Sigma_b, \sigma_{err}\}$ ▷ Parameter Vector
2. Define Objective (JIT Compiled)
function $\text{Loss}(\theta, L_{obs})$
    Unpack $\mu_p, \mu_b, \Sigma_p, \Sigma_b$ from θ
    $\mu_{ij} \leftarrow \mu_{p,i} - \mu_{b,j}$ ▷ Difference Mean
    $\Sigma_{ij} \leftarrow \Sigma_{p,i} + \Sigma_{b,j}$ ▷ Combined Uncertainty
    // Closed-Form Expected Attribution (MGF)
    $Term_1 \leftarrow |I + 2\Sigma_{ij}|^{-1/2}$
    $Term_2 \leftarrow \exp(-\mu_{ij}^\top (I + 2\Sigma_{ij})^{-1} \mu_{ij})$
    $\hat{L}_{ij} \leftarrow Term_1 \cdot Term_2$
    // Negative Log-Likelihood (Gaussian Error)
    $\mathcal{J} \leftarrow \sum_{i,j} \left(\frac{(L_{obs,ij} - \hat{L}_{ij})^2}{2\sigma_{err}^2} + \log(\sigma_{err}) \right)$
    return $\mathcal{J}$
end function
3. Optimization Loop (Adam + AutoDiff)
$UpdateFn \leftarrow \text{JIT}(\nabla_\theta \text{Loss})$ ▷ Compile Gradient Function
while $\Delta\mathcal{J} > \text{tol}$ **do**
    $g \leftarrow UpdateFn(\theta, L)$
    $\theta \leftarrow \text{Adam}(\theta, g, lr = 0.01)$
end while
return Optimized θ

B.4 Hyperparameter Configuration

To ensure reproducibility of the Stage 1 DML residuals, Table 3 lists the exact search space used in the `RandomizedSearchCV` procedure for the Gradient Boosting regressors.

Table 3: Hyperparameter Grid ($\mathcal{G}$) for Confounding Models

Parameter	**Search Space**
Max Depth	$[2, 3, 4]$
Min Samples Leaf	$[3, 5, 10, 15]$
Subsample Ratio	$[0.5, 0.7, 0.9]$
Learning Rate	$[0.01, 0.05, 0.1]$
N Estimators	$[100, 200, 300]$

C Mathematical Discussion

C.1 Theoretical Foundation: The Frisch-Waugh-Lovell (FWL) Theorem

In Stage 1, we rely on the Double Machine Learning (DML) framework to estimate the parameter γ. The validity of this approach rests on the Frisch-Waugh-Lovell (FWL) theorem, which we derive here to demonstrate how it eliminates confounding.
Consider the partially linear structural model:

$$Y = \gamma Z + f(X) + \epsilon, \quad E[\epsilon|X, Z] = 0 \tag{13}$$

where Z is the treatment (bias factor) and $f(X)$ is a complex, non-linear confounding function of performance metrics. A direct regression of Y on Z would be biased because Z is correlated with X (e.g., Draft Number correlates with Performance).
To isolate γ, we first take the conditional expectation of the equation with respect to the confounders X:

$$\begin{aligned} E[Y|X] &= E[\gamma Z + f(X) + \epsilon \mid X] \\ E[Y|X] &= \gamma E[Z|X] + f(X) + E[\epsilon \mid X] \end{aligned} \tag{14}$$

Note that since $f(X)$ is a constant given X, it remains unchanged.
Next, we subtract equation (14) from the original structural equation:

$$\begin{aligned} Y - E[Y|X] &= (\gamma Z + f(X) + \epsilon) - (\gamma E[Z|X] + f(X)) \\ Y - E[Y|X] &= \gamma(Z - E[Z|X]) + \epsilon \end{aligned} \tag{15}$$

Critically, the unknown confounding function $f(X)$ cancels out completely.
We define the residuals (the "unexplained" components) as:

$$\tilde{Y} = Y - E[Y|X] \quad \text{and} \quad \tilde{Z} = Z - E[Z|X] \tag{16}$$

Substituting these back yields a simplified linear equation free of confounding:

$$\tilde{Y} = \gamma \tilde{Z} + \epsilon \tag{17}$$

This derivation proves that γ can be recovered by a simple OLS regression of the outcome residuals $\tilde{Y}$ on the treatment residuals $\tilde{Z}$, provided that the conditional expectations $E[Y|X]$ and $E[Z|X]$ are estimated consistently (which we achieve via Gradient Boosting in Algorithm 1).

C.2 Mathematical Derivation: The Probabilistic Integral

In Stage 3, we assert that the expected attribution $\hat{L}_{ij}$ has a closed-form solution, allowing for efficient gradient-based optimization. Here we provide the formal derivation.
We define the similarity between player p and factor b as the Gaussian kernel of their difference vector $\delta \sim \mathcal{N}(\mu, \Sigma)$, where $\mu = \mu_p - \mu_b$ and $\Sigma = \Sigma_p + \Sigma_b$. The objective is to calculate the expectation of the similarity kernel $g(\delta) = \exp(-\delta^\top \delta)$:

$$\hat{L}_{ij} = \mathbb{E}_\delta[g(\delta)] = \int_{\mathbb{R}^d} \exp(-\delta^\top \delta) \cdot \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(\delta - \mu)^\top \Sigma^{-1}(\delta - \mu)\right) d\delta \tag{18}$$

This integral represents the expectation of a quadratic form. We utilize the property of the Moment Generating Function (MGF) for a quadratic Gaussian variable $Q = \delta^\top A\delta$. The MGF is defined as $M_Q(t) = \mathbb{E}[e^{tQ}]$.
For a general Gaussian $x \sim \mathcal{N}(\mu, \Sigma)$ and symmetric matrix A, the MGF is given by the standard identity:

$$\mathbb{E}[\exp(tx^\top Ax)] = |I - 2t\Sigma A|^{-1/2} \exp\left(-\frac{1}{2}\mu^\top \Sigma^{-1}\left[I - (I - 2t\Sigma A)^{-1}\right]\mu\right) \tag{19}$$

To map our problem to this identity, we set the quadratic matrix $A = I$ (the identity matrix) and the scalar parameter $t = -1$:

$$\mathbb{E}[\exp(-\delta^\top \delta)] = |I + 2\Sigma|^{-1/2} \exp\left(-\frac{1}{2}\mu^\top \Sigma^{-1}\left[I - (I + 2\Sigma)^{-1}\right]\mu\right) \tag{20}$$

The exponential term can be simplified using the matrix identity $B^{-1} - (B + C)^{-1} = B^{-1}C(B + C)^{-1}$. Letting $B = \Sigma^{-1}$ and using the fact that $(I + 2\Sigma)^{-1} = (\Sigma(\Sigma^{-1} + 2I))^{-1}$, the term inside the brackets simplifies algebraically to:

$$\Sigma^{-1}\left[I - (I + 2\Sigma)^{-1}\right] = 2(I + 2\Sigma)^{-1} \tag{21}$$

Substituting this back into the exponential term cancels the $-\frac{1}{2}$ factor, yielding the final closed-form solution used in our JAX kernel:

$$\hat{L}_{ij} = \underbrace{|I + 2\Sigma|^{-1/2}}_{\text{Uncertainty Penalty}} \cdot \underbrace{\exp\left(-\mu^\top (I + 2\Sigma)^{-1}\mu\right)}_{\text{Distance Penalty}} \tag{22}$$

This algebraic solution transforms an intractable high-dimensional integration into a sequence of standard linear algebra operations (determinants and inverses), which are fully differentiable and parallelizable on GPU accelerators.

C.3 Theoretical Justification: Probabilistic vs. Deterministic Embedding

Our choice to model players and bias factors as Gaussian distributions ("clouds") rather than fixed coordinates ("points") offers a distinct theoretical advantage over traditional Multi-Dimensional Scaling (MDS).
In a deterministic framework (e.g., standard Stress Minimization), the objective function seeks to minimize the difference between observed dissimilarities δ_{ij} and geometric distances $d_{ij}(\mathbf{x})$:

$$\mathcal{L}_{MDS} = \sum_{i,j} (\delta_{ij} - d_{ij}(\mathbf{x}))^2 \tag{23}$$

This approach treats all data points with equal confidence. If the DML residuals for a player are noisy (high variance), a deterministic model will still force an exact coordinate placement to minimize the squared error, often resulting in overfitting and a misleadingly precise map.
In contrast, our probabilistic formulation incorporates an explicit uncertainty term via the variance Σ. The expected attribution $\hat{L}_{ij}$ derived in Section C.2 contains a critical regularization mechanism:

$$\hat{L}_{ij} \propto \underbrace{|I + 2\Sigma|^{-1/2}}_{\text{Volume Penalty}} \exp(\dots) \tag{24}$$

This determinant term acts as an *adaptive regularizer*.
Crucially, this variance term handles the information loss inherent in projecting high-dimensional data (10 bias factors) into a low-dimensional space (3D). A player might possess a complex bias profile that is geometrically impossible to satisfy perfectly in 3D (e.g., they are strongly attracted to two bias factors that are positioned far apart due to the global topology).

- A *deterministic* model would place this player in the geometric average, satisfying neither constraint well, yet presenting the result as a precise point.
- Our *probabilistic* model resolves this geometric contradiction by increasing the player's variance Σ. The model effectively "smears" the player's location across the space between the conflicting factors.

Therefore, high variance in our map does not just represent statistical noise; it encodes the *lack of fit* caused by dimension reduction. This ensures the visualization is honest: precise points represent players with clear, structurally consistent bias profiles, while diffuse clouds represent players with complex or contradictory profiles that defy simple low-dimensional categorization.

D Collaboration Log (Selected Git History)

Evidence of our sustained project scope is documented via our GitHub repository history. The following table summarizes key technical milestones and contributions from the initial project setup (6 weeks ago) to final deployment.

Time	Contributor	Key Commit / Milestone
6 weeks ago	T. Venner	Initial commit & Project Plan setup
6 weeks ago	T. Venner	Implemented team-specific endpoint scraping
5 weeks ago	M. Chen	Added salary scraping script (`get_salary.py`)
5 weeks ago	ggg808	Added NBA player performance stats collection
5 weeks ago	L. Garibay-Estrada	Implemented helper functions for data merging
5 weeks ago	M. Chen	Scraped salary cap data into `raw_salary_caps.csv`
4 weeks ago	T. Venner	Created merged dataset & feature engineering
3 weeks ago	L. Garibay-Estrada	Added `train_f_model` (Salary $\sim$ Performance) module
3 weeks ago	M. Chen	Implemented pipeline for treatment models ($\hat{h}$)
3 weeks ago	ggg808	Completed DML bias analysis Module 4
3 weeks ago	T. Venner	Validated full orchestration pipeline (`main.py`)
2 weeks ago	T. Venner	Added JOS (Years of Service) & Contract Type engineering
2 weeks ago	ggg808	Updated Streamlit visualizations for DML results
13 days ago	T. Venner	Implemented JAX-based MLE Fitter for 3D Map
13 days ago	T. Venner	Added DML output mapping to Attribution Matrix (L)
12 days ago	M. Chen	Organized visuals into Streamlit pages
4 days ago	T. Venner	Updated gradient boosting methods for $\hat{f}$ and $\hat{h}$ models
4 days ago	T. Venner	Added metric framework (Log Loss, R2) for performance tracking
3 days ago	T. Venner	Tuned hyperparameters; achieved optimal R^2 for baseline
6 hours ago	L. Garibay-Estrada	Final Streamlit page updates
4 hours ago	M. Chen	Updated website to include bias vs. performance plots
1 hour ago	T. Venner	Final full pipeline run
15 minutes ago	T. Venner	Finished writing report. Compiled and uploaded

Note: This log filters out minor commits (e.g., typos, merges) to highlight substantive technical contributions.

E Supplementary Data Tables

E.1 Variable Dictionary Feature Selection

Table 5 details the variables used in our modeling pipeline. Note that initial feature selection was performed using a preliminary Gradient Boosting regressor; features with an importance score of exactly 0.00 (e.g., `PNR_PPP`, `OREB_PCT`) were removed from the final X vector to maximize validation R^2.

Table 5: Variable Descriptions and Transformations

Variable	Type	Description
Outcome Variable (Y)		
Salary	Continuous	Total annual contract value (Log-transformed for modeling).
Performance Metrics (X) - Source: NBA API		
OFF_RATING	Rate	Points produced per 100 possessions.
DEF_RATING	Rate	Points allowed per 100 possessions.
NET_RATING	Rate	Point differential per 100 possessions.
CLUTCH_PTS	Count	Points scored in "Clutch" time (last 5 min, score $\leq$ 5 pts).
CLUTCH_GP	Count	Number of games involving clutch time.

Continued on next page...

Variable	Type	Description
PIE	Index	Player Impact Estimate (NBA.com holistic metric).
USG_PCT	Rate	Percentage of team plays used by player while on court.
TS_PCT	Rate	True Shooting % (adjusts for 3-pointers and Free Throws).
EFG_PCT	Rate	Effective Field Goal %.
AST_TO	Ratio	Assist-to-Turnover ratio.
AST_RATIO	Rate	Assists per 100 possessions.
REB_PCT	Rate	Percentage of available rebounds grabbed.
TM_TOV_PCT	Rate	Percentage of team possessions ending in a turnover.
PACE	Rate	Possessions per 48 minutes.
POSS	Count	Total possessions played.
FGM_PG / FGA_PG	Rate	Field Goals Made / Attempted per Game.
GP	Count	Games Played.
MIN	Count	Total Minutes Played.
AVG_SPEED	Rate	Average speed on court (mph).
DIST_MILES	Count	Total distance traveled on court.
ISO_PTS	Count	Points scored in Isolation plays.
POST_PTS	Count	Points scored in Post-up plays.
RIM_DFG_PCT	Rate	Opponent FG% at the rim when defended by player.
Bias Factors (Z) - Source: Spotrac / Instagram / Wiki		
Age	Continuous	Player age in years (derived from Birthdate).
DRAFT_NUMBER	Discrete	Overall pick number (1-60). Undrafted imputed as 61.
Followers	Continuous	Instagram followers (Log-transformed).
is_USA	Binary	1 if Country is USA, 0 otherwise.
active_cap	Continuous	Team's total active salary cap allocation.
dead_cap	Continuous	Team's dead money (salary paid to cut players).
OWNER_NET_W...	Continuous	Owner Net Worth in Billions (Log-transformed).
Capacity	Discrete	Stadium seating capacity.
STADIUM_COST	Continuous	Construction cost of home arena.
STADIUM_YEAR...	Discrete	Year the home arena opened.

E.2 Full Model Diagnostics

Table 6 presents the complete feature importance ranking for the Stage 1 Outcome Model ($\hat{f}$).

Table 6: Full Feature Importance (Gradient Boosting - MDI)

Feature	Importance	Feature	Importance
CLUTCH_PTS	0.1843	POSS	0.0352
NET_RATING	0.0914	TS_PCT	0.0329
MIN	0.0898	USG_PCT	0.0324
CLUTCH_GP	0.0795	OFF_RATING	0.0288
FGM_PG	0.0746	PACE	0.0247
FGA_PG	0.0690	PIE	0.0215
GP	0.0567	POST_PTS	0.0194
AVG_SPEED	0.0471	DIST_MILES	0.0193

Note: Only top 16 features shown. Features with 0.0 importance were pruned.

Table 7 details the performance of all nuisance models used in the DML pipeline. The low R^2 values for bias factors (e.g., Age, Draft Number) confirm that these attributes are orthogonal to on-court performance, satisfying the DML structural assumption.

Technical Note on Negative R^2: The reader may observe negative R^2 values for factors such as Age and Stadium Cost. In out-of-sample cross-validation, a negative R^2 indicates that the trained model prediction had a higher Mean Squared Error than a naive baseline (simply predicting the global mean). In the context of DML, this is a *positive diagnostic result*. It empirically confirms that these contextual factors are effectively orthogonal to on-court performance metrics. The Gradient Boosting model attempted to find a predictive signal in X but found only noise; consequently, the "residual" is essentially the original variable centered around its mean, confirming that no non-linear confounding exists for these specific factors.

Table 7: Snapshot of Nuisance Model Performance (5-Fold CV)

Target Variable	**Type**	R^2 **(Mean)**	**RMSE**	**Log Loss**
Salary_Model ($\hat{f}$)	Regression	**0.4836**	1.074	-
Age	Regression	-0.0028	3.986	-
Capacity	Regression	-0.1483	1117.6	-
DRAFT_NUMBER	Regression	-0.1143	18.395	-
Followers	Regression	0.0369	4.243	-
OWNER_NET_WORTH	Regression	-0.1617	1.200	-
STADIUM_COST	Regression	-0.2889	4.58×10^8	-
STADIUM_YEAR	Regression	-0.0048	11.458	-
active_cap	Regression	0.0693	1.52×10^7	-
dead_cap	Regression	-0.2841	9.81×10^6	-
is_USA	Classification	-	-	0.6549

E.3 Complete Market Price Estimates

Table 8 presents the full output of the final debiased regression ($\epsilon_Y \sim \epsilon_Z$), sorted by statistical significance.

Table 8: Learned Market Prices ($\hat{\gamma}$) - Full Results

Bias Factor	**Price** ($\hat{\gamma}$)	**Std. Error**	**P-Value**
Age	**0.08014**	0.02100	**0.00014**
Followers	**0.04317**	0.01890	**0.02238**
active_cap	-0.00000	0.00000	0.04865
is_USA	-0.20879	0.12626	0.09821
STADIUM_YEAR_OPENED	-0.01317	0.00939	0.16085
Capacity	0.00006	0.00006	0.30503
OWNER_NET_WORTH_B	-0.06671	0.06766	0.32417
DRAFT_NUMBER	0.00160	0.00407	0.69468
dead_cap	-0.00000	0.00000	0.95302
STADIUM_COST	0.00000	0.00000	0.99416
Constant	-0.01390	0.06618	0.83361

F Deployment Interface

The following screenshot captures the "Structural Bias Score" visualization from our Streamlit web application. This interface allows users to select a specific player and view their specific attribution profile (L_i) against the league average.