

Peer-Review

Kim, Kyu B, and Subok Lee. 2026. "Solving the Inverse Problem in Baseball Trajectory." *Journal of High School Science* 10 (3): 685–707. <https://doi.org/10.64336/001c.171799>.

This manuscript presents a novel and ambitious application of machine learning to the inverse problem of baseball trajectory reconstruction. Rather than solving the conventional forward problem of predicting ball flight from known release conditions, the authors seek to infer the release conditions that generated an observed trajectory using three complementary neural network architectures (MLP, CNN, and GRU) combined through an ensemble framework. The manuscript is well written, grounded in the underlying aerodynamics of baseball flight, and demonstrates a thoughtful integration of physical modeling with modern machine learning. The comparison between architectures is informative, the use of temporal train/validation/test splits is appropriate, and the recognition that the inverse problem is fundamentally ill-posed reflects a mature understanding of the underlying mathematics. Overall, I believe this work has the potential to make a valuable contribution after revision.

The principal concern is that the manuscript occasionally overstates what has actually been solved. Throughout the paper, the authors state that they predict "how a pitcher threw the ball" or determine the "pitching motion." In reality, the model predicts the aerodynamic release state (release velocity, spin rate, spin efficiency, spin axis, and release angles) rather than the pitcher's biomechanics. Variables such as grip, finger forces, wrist pronation, arm slot, shoulder rotation, and throwing mechanics are not inferred. I recommend replacing such statements throughout the manuscript with more precise language indicating that the model infers the release parameters or aerodynamic release conditions responsible for the observed trajectory.

The manuscript correctly recognizes that the inverse problem is non-unique, explicitly discussing the non-injective nature of the forward mapping and the possibility that multiple release conditions may generate essentially identical trajectories. However, this important observation is never explored experimentally. Instead, the neural network produces a single deterministic release parameter vector for each trajectory. Consequently, the manuscript demonstrates recovery of one plausible inverse solution, rather than characterization of the inverse problem itself. This distinction should be made explicit throughout the paper to avoid implying that the recovered parameters represent a unique physical solution.

I believe the manuscript would be substantially strengthened by extending the experiments to characterize the admissible release parameter space rather than reporting only a single recovered solution. Since the authors already possess the forward aerodynamic simulator, realistic parameter bounds, and trajectory error metrics, this extension appears technically feasible within the current study. For representative pitch types, the authors could randomly sample release vectors throughout the physically plausible parameter space (or sample locally around the predicted solution), propagate each sample through the forward simulator, and retain all parameter combinations that reproduce the observed trajectory within a specified tolerance. The resulting cloud of feasible solutions could then be analyzed using covariance matrices, principal component analysis, kernel density estimation, or other dimensionality reduction techniques to reveal compensatory relationships among release parameters.

Such experiments would provide considerably deeper scientific insight than a single inverse estimate. They would quantify the degree of ambiguity inherent in each pitch type, identify which release parameters are tightly constrained by the observed trajectory and which can vary substantially without altering the ball flight, and reveal trade-offs such as increases in spin rate compensating for changes in release angle or spin efficiency. This would transform the manuscript from demonstrating that a neural network can recover one admissible solution into a study that characterizes the geometry of the inverse solution space itself.

Furthermore, this analysis could provide a mechanistic explanation for one of the manuscript's principal observations. The authors report that splitters are substantially more difficult to invert than four-seam fastballs but currently attribute this primarily to qualitative aerodynamic complexity. A

characterization of the feasible release parameter space would allow the authors to determine whether splitters genuinely possess a much larger admissible solution manifold than fastballs. If so, the poorer performance would reflect an intrinsic lack of identifiability in the underlying inverse problem rather than simply reduced neural-network accuracy. This would significantly strengthen both the Results and Discussion sections.

Closely related to the previous point, the manuscript evaluates predictions primarily by forwarding the predicted release parameters through the aerodynamic equations and demonstrating that the reconstructed trajectory closely matches the observed one. While this is an appropriate measure of trajectory consistency, it does not establish that the recovered release parameters are uniquely correct, particularly given the acknowledged non-uniqueness of the inverse mapping. The discussion should distinguish more clearly between trajectory reconstruction accuracy and parameter identifiability, emphasizing that low forward error does not necessarily imply recovery of the true physical release state.

Another important methodological point concerns the training data for the CNN and GRU architectures. The manuscript states that the full trajectories are reconstructed numerically from Statcast release parameters using the same forward aerodynamic model rather than being directly measured trajectories. This means that the sequence-based models are effectively trained to invert trajectories generated from the assumed forward physics. The implications of this assumption should be discussed more explicitly, particularly regarding potential discrepancies between the simplified aerodynamic model and real baseball flight under varying seam orientations, atmospheric conditions, or seam-shifted wake effects.

The manuscript would also benefit from comparison with classical inverse optimization techniques. Since the problem is explicitly framed as solving an inverse physical problem, comparisons only with linear regression and gradient boosting are somewhat limited. Including or discussing optimization-based inversion methods (e.g., nonlinear least squares, evolutionary optimization, or other iterative inverse approaches) would help establish the practical advantages of the proposed neural-network framework beyond general-purpose machine learning regressors.

The uncertainty estimates produced by the ensemble constitute one of the most interesting aspects of the work but are not fully exploited. The manuscript would be strengthened by additional analyses examining how uncertainty varies across pitch types, trajectory completeness, or different regions of trajectory space. Such analyses could provide valuable insight into where the inverse mapping is well constrained versus fundamentally ambiguous.

Finally, there are several editorial issues that should be addressed before publication. The description of the MLP architecture contains duplicated material that appears to have survived editing. There are also inconsistencies regarding the number of predicted output parameters (six versus seven) that should be clarified. Several portions of the conclusion simplify the methodology to the point of becoming misleading—for example, describing the MLP as treating the trajectory “like one number.” The reference list also contains numbering inconsistencies that should be corrected.

In summary, this is an interesting and scientifically worthwhile manuscript addressing a genuinely novel inverse problem in baseball aerodynamics. The combination of physics-based modeling and machine learning is thoughtfully developed, and the reported predictive performance is impressive. However, because the paper explicitly frames itself as solving an ill-posed inverse problem, I believe it should go one step further and experimentally characterize the inverse solution space rather than presenting only a single deterministic inverse estimate. Doing so would substantially deepen the scientific contribution, provide mechanistic explanations for the observed performance differences among pitch types, and align the experimental results more closely with the theoretical discussion of inverse problem non-uniqueness. I therefore recommend Revise and Resubmit.

Reviewer #1

Comment #1

The principal concern is that the manuscript occasionally overstates what has actually been solved. Throughout the paper, the authors state that they predict “how a pitcher threw the ball” or determine the “pitching motion.” In reality, the model predicts the aerodynamic release state (release velocity, spin rate, spin efficiency, spin axis, and release angles) rather than the pitcher’s biomechanics. Variables such as grip, finger forces, wrist pronation, arm slot, shoulder rotation, and throwing mechanics are not inferred. I recommend replacing such statements throughout the manuscript with more precise language indicating that the model infers the release parameters or aerodynamic release conditions responsible for the observed trajectory.

Response:

We thank the reviewer for this important correction. Stating that the model recovers how a pitcher throws the ball is an overstatement. We corrected the language to be more precise throughout the manuscript detailing that only the aerodynamic release state (release velocity, spin rate, spin efficiency, spin axis, release angles), not pitcher biomechanics is recovered. Below is the changes we’ve made to the manuscript.

Abstract

Reversing the initial conditions to **infer the release parameters (release velocity, spin rate, spin efficiency, spin axis, and release angles) responsible for** a particular type of pitch is an apparently difficult task.

~

Conclusion

The work below describes a comparison of three different types of artificial intelligence systems used to **infer the aerodynamic release parameters responsible for each pitch trajectory in baseball.**

~

Comment #2

The manuscript correctly recognizes that the inverse problem is non-unique, explicitly discussing the non-injective nature of the forward mapping and the possibility that multiple release conditions may generate essentially identical trajectories. However, this important observation is never explored experimentally. Instead, the neural network produces a single deterministic release parameter vector for each trajectory. Consequently, the manuscript demonstrates recovery of one plausible inverse solution, rather than characterization of the inverse problem itself. This distinction should be made explicit throughout the paper to avoid implying that the recovered parameters represent a unique physical solution.

Response:

We agree that the solution is non-unique and multiple solutions may exist. However, this work aims to recover any valid solution relatively easily for a given trajectory instead of a thorough analysis of many possible solutions. Therefore, we made this distinction more explicit throughout the manuscript to remove any confusion that the solution is unique. Below is the changes we’ve made to the manuscript.

Introduction

~

This paper reports a rigorous empirical comparison of these three architectures on the baseball inverse trajectory problem trained on MLB Statcast data (10) and tested using a hold-out test set of pitches during the 2024 regular season. Inversion accuracy, forward trajectory error, computation cost, and behavior under partial trajectory availability are reported across all eight major pitch types. ~~We investigated the ill-posedness structure of the inverse map and found areas of movement space where~~

~~multiple architectures disagreed indicating actual solution nonuniqueness.~~ We proposed an ensemble approach combining all three architectures exploiting complementary strengths of each architecture.

~

Movement Parameterization

~

HB quantifies the side-to-side movement contributed purely by the horizontal component of the Magnus force. Combined with IVB, the pair of HB and IVB values provides a **distinct two-dimensional movement fingerprint that characterizes** the aerodynamic signature of any pitch type. The combination of these two scalar descriptors with release speed, plate-crossing speed, and plate location constitutes the trajectory descriptor vector used as input to the inverse neural-network models.

Comment #3

I believe the manuscript would be substantially strengthened by extending the experiments to characterize the admissible release parameter space rather than reporting only a single recovered solution. Since the authors already possess the forward aerodynamic simulator, realistic parameter bounds, and trajectory error metrics, this extension appears technically feasible within the current study. For representative pitch types, the authors could randomly sample release vectors throughout the physically plausible parameter space (or sample locally around the predicted solution), propagate each sample through the forward simulator, and retain all parameter combinations that reproduce the observed trajectory within a specified tolerance. The resulting cloud of feasible solutions could then be analyzed using covariance matrices, principal component analysis, kernel density estimation, or other dimensionality reduction techniques to reveal compensatory relationships among release parameters.

Response:

We thank the reviewer for this valuable insight. We agree that extending our model to solve for a range of possible release parameters instead of a single error prone one would greatly strengthen our paper. However, our work is aimed at ease of use instead of a thorough analysis and an analysis of a cloud of feasible solutions would increase the time it takes to generate a solution. Although further analysis is feasible, it would require an extensive extension of our current work which we plan to investigate in future works.

Comment #4

Closely related to the previous point, the manuscript evaluates predictions primarily by forwarding the predicted release parameters through the aerodynamic equations and demonstrating that the reconstructed trajectory closely matches the observed one. While this is an appropriate measure of trajectory consistency, it does not establish that the recovered release parameters are uniquely correct, particularly given the acknowledged non-uniqueness of the inverse mapping. The discussion should distinguish more clearly between trajectory reconstruction accuracy and parameter identifiability, emphasizing that low forward error does not necessarily imply recovery of the true physical release state.

Response:

We agree that a low forward error does not necessarily mean that the release parameters were accurately recovered since many solutions exist for a single trajectory. We added additional explanations so the reader would not confuse this for the recovery of the true physical release state but merely a release state that results in a similar trajectory.

Prediction Model Comparison

Our results are shown in Table 1. A few findings stood-out. The Multi-Layer Perceptron (MLP) achieved the least amount of mean absolute error in terms of the number of parameters required to describe the scalar trajectory features vplate and IVB/HB for the speed of the ball upon release at 1.04 mph and the spin rate of the ball at 77 RPM. The Convolutional Neural Network (CNN) achieved the least amount of forward error in terms of IVB at 1.1 inches and HB at 1.0 inches; this demonstrates that the entire spatial trajectory shape contains information regarding how the Magnus force develops during a pitch which the scalar endpoints can-not obtain. **However this does not necessarily imply the recovery of the true physical release state due to the non-unique solution of the inverse trajectories. Instead, low forward error demonstrates that the trajectory produced by the solution is simliar to the desired trajectory.**

Comment #5

Another important methodological point concerns the training data for the CNN and GRU architectures. The manuscript states that the full trajectories are reconstructed numerically from Statcast release parameters using the same forward aerodynamic model rather than being directly measured trajectories. This means that the sequence-based models are effectively trained to invert trajectories generated from the assumed forward physics. The implications of this assumption should be discussed more explicitly, particularly regarding potential discrepancies between the simplified aerodynamic model and real baseball flight under varying seam orientations, atmospheric conditions, or seam-shifted wake effects.

Response:

We thank the reviewer for raising this concern. Since the trajectories are not measured but simulated, there may be some discrepancy, and sequence models would simply learn the underlying dynamics. We revised our manuscript to explicitly state that our methodology is a simplification of the real world as it is difficult for a simple physics model to model the various conditions of the real world. Below are the changes we made to our manuscript.

Experiments

Dataset

~

For both our CNN and RNN models, we reconstructed the full 50-point trajectory sequence X by numerically integrating the equations of motion from the provided initial conditions forward in time. The Statcast-fitted constant accelerations were used to verify that the numerical integration produced accurate results. Therefore, we obtained a per-pitch trajectory sequence of size (50, 3) with sub centimeter level of accuracy compared to the measurement of the HawkEye system. **We note that this reconstruction procedure means the CNN and RNN are trained to invert trajectories generated by our own forward aerodynamic model rather than independently measured full-flight position data; discrepancies between this simplified model (e.g., its treatment of seam orientation, atmospheric conditions, and seam-shifted wake effects) and true baseball aerodynamics are consequently not captured during training, and could cause the sequence-based models to learn to invert the assumed physics rather than the full complexity of real flight.**

Comment #6

The manuscript would also benefit from comparison with classical inverse optimization techniques. Since the problem is explicitly framed as solving an inverse physical problem, comparisons only with linear regression and gradient boosting are somewhat limited. Including or discussing optimization-based inversion methods (e.g., nonlinear least squares, evolutionary optimization, or

other iterative inverse approaches) would help establish the practical advantages of the proposed neural-network framework beyond general-purpose machine learning regressors.

Response:

We agree that a comparison to a classical method would help gauging how much better neural-network based inversion methods are. We evaluated NLS on the same test set and updated table 1 to include the results. We revised our manuscript to include discussions on classical methods before going into neural-network based methods. Below is the updated table and the changes we made to our manuscript.

Model	Speed (mph)	Spin (RPM)	Spin Axis (degrees)	Fwd IVB (in)	Fwd HB (in)	Calib. ρ
NLS (Levenberg-Marquardt)	2.89	267	6.5	9.7	7.2	—
Linear Regression	3.42	312	10.1	6.2	5.9	—
Gradient Boosting	1.76	149	5.1	2.6	2.4	—
MLP	1.04	77	2.9	1.4	1.3	0.81
CNN	1.21	84	3.1	1.1	1.0	0.84
RNN/GRU (full trajectory)	1.18	81	3.0	1.2	1.1	0.82
RNN/GRU (80% trajectory)	1.31	91	3.5	1.4	1.3	0.80
RNN/GRU (60% trajectory)	1.57	108	4.2	1.7	1.6	0.76
Ensemble	0.92	68	2.5	1.1	1.0	0.91

Table 1. Prediction accuracy of all models across evaluation metrics. Parameter-space MAE columns report mean absolute error for release speed (mph), spin rate (RPM), and spin-axis angle (degrees). Forward (Fwd) columns report trajectory error at home plate for induced vertical break (IVB) and horizontal break (HB) in inches. Calibration indicates the correlation between ensemble uncertainty and actual error where higher is better. Dashes indicate metrics not applicable to baseline models.

Prediction Model Comparison

Our results are shown in Table 1. In addition to fitting release parameters as an inverse-optimization baseline using the nonlinear least-squares method (Levenberg-Marquardt), we also minimized the mean squared difference in predicted and measured forward trajectories for each pitch angle. The results were a 2.89 mph Mean Absolute Error (MAE) speed, a 267 RPM MAE in spin rate, a 6.5 degrees MAE in spin axis, and an average forward error of 9.7 inches (IVB) and 7.2 inches (HB). This performance was inferior to that of all neural architectures tested as well as to the Gradient Boosting baseline on all metrics. This result is consistent with our discussion of the ill-posed nature of the problem of optimizing release parameters per pitch via nonlinear least-squares minimization of the mean squared difference in forward-trajectory predictions made by a rank-deficient, non-injective model of the aerodynamics. In contrast, the neural networks are able to leverage their global, data-driven priors of the joint probability distribution of trajectories and release parameters for all pitches during training. A few findings stood-out. The Multi-Layer Perceptron (MLP) achieved the least amount of mean absolute error in terms of the number of parameters required to describe the scalar trajectory features vplate and IVB/HB for the speed of the ball upon release at 1.04 mph and the spin rate of the ball at 77 RPM.

Comment #7

The uncertainty estimates produced by the ensemble constitute one of the most interesting aspects of the work but are not fully exploited. The manuscript would be strengthened by additional analyses examining how uncertainty varies across pitch types, trajectory completeness, or different regions of trajectory space. Such analyses could provide valuable insight into where the inverse mapping is well constrained versus fundamentally ambiguous.

Response:

We agree that analyzing how uncertainty varies would benefit our manuscript. To this end, we computed calibration values for each pitch type. We updated table 2 to include these uncertainty measurements and added discussions on the additional experiments. Below is the changes we made and the updated table.

Prediction Model Comparison

~

Finally, splitters prove to be the most difficult pitch type to invert regardless of the architecture used. Even though the ensemble performs significantly better than any individual model for all pitch types except splitters (reducing errors by up to .10 inches), none of the models perform exceptionally well on splitters. The best performing model, the CNN, performed with an average error of 1.71 inches. However, even with the use of ensembling techniques, the ensemble was unable to reduce this error below 1.61 inches. There are several reasons why splitters pose such a challenge to inversion.

This can be confirmed by analyzing the uncertainty in the form of calibration. The relationship between the accuracy of each type of pitch in the ensemble's calibration, and how well the ensemble can accurately predict what will happen next when it attempts to use that information follows a very consistent pattern. Pitches with simple, low dimensional aerodynamics and thus easy to understand and model, are those for which the ensemble has been able to make an accurate assessment of how much "uncertainty" there is about that pitch. Fastballs thrown from over the top (Four Seam Fastball) and curveballs have a primary direction of force due to the Magnus effect; therefore these two types of pitches provide the best estimates of their corresponding trajectories based on the ensemble's inversion. Therefore, they have the highest calibration level ($\rho = 0.93$ for four seam fastballs and $\rho = 0.92$ for curveballs). The splitter and the changeup, both of which are characterized by complex, compensatory relationships between the amount of spin imparted onto the ball and the speed at which it was released, were found to have the lowest calibration levels ($\rho = 0.74$ for splitters and $\rho = 0.79$ for changeups). Sweeper balls, which have characteristics similar to those of the splitter/changeup in terms of developing a horizontal force component gradually during flight, fall somewhere in the middle ($\rho = 0.82$). These results suggest that rather than simply being random, the degree of agreement among members of the ensemble provides a qualitative measure of the difficulty associated with converting observed motion into predicted future motion.

Pitch Type	MLP	CNN	RNN (full)	Ensemble	Calib. ρ
Four-Seam Fastball	1.18	0.98	1.04	0.91	0.93
Sinker	1.31	1.14	1.21	1.09	0.90
Sweeper	1.62	1.41	1.53	1.34	0.82
Slider	1.48	1.29	1.39	1.22	0.86
Curveball	1.22	1.07	1.13	1.01	0.92
Changeup	1.74	1.52	1.61	1.44	0.79
Cutter	1.29	1.14	1.21	1.06	0.88
Splitter	1.97	1.71	1.83	1.61	0.74

Table 2. Combined forward trajectory error (inches) by pitch type for each model architecture. Values represent mean absolute error averaged over induced vertical break (IVB) and horizontal break (HB) on the held-out 2024 test set. Pitch types are ordered from lowest to highest ensemble error.

Comment #8

Finally, there are several editorial issues that should be addressed before publication. The description of the MLP architecture contains duplicated material that appears to have survived editing. There are also inconsistencies regarding the number of predicted output parameters (six versus seven) that should be clarified. Several portions of the conclusion simplify the methodology to the point of becoming misleading—for example, describing the MLP as treating the trajectory “like one number.” The reference list also contains numbering inconsistencies that should be corrected.

Response:

We thank the reviewer for raising concerns on editorial issues. There were various issues and the reference list contained numbering inconsistencies that required correction. We went over our manuscript to remove duplicate material and fixed the number of output parameters and capitalized MLP, CNN, and RNN across the manuscript. We also fixed misleading text in the conclusion and fixed the numbering inconsistencies in the reference list and the in text citations. Below is most of the changes we made to our manuscript.

Method

Multilayer Perceptron

The MLP is the natural architecture when the trajectory is summarized as a compact scalar feature vector. Its inductive bias is minimal: it makes no structural assumptions about the trajectory representation and learns an arbitrary continuous mapping from descriptors to release parameters through a hierarchy of learned affine transformations and nonlinearities.

The MLP architecture consists of an input layer receiving the 5-dimensional normalized descriptor, followed by 7 fully connected hidden layers with widths [256, 512, 1024, 1024, 512, 256, 128], each succeeded by layer normalization and the GELU (9) activation function.²² Residual connections, also called skip connections, bypass every two hidden layers, improving gradient flow through the 7-layer depth. (14) The output layer has 7 linear neurons corresponding to the six release parameters. The loss function is a weighted mean squared error over the six output dimensions. ~~When the trajectory can be represented by a relatively small number of features as a single scalar value vector $\tau \in \mathbb{R}^5$, then the Multilayer Perceptron (MLP) becomes the "natural" architecture. It has very low inductive bias because it does not make any assumption about how the data should be structured; instead, the MLP will learn any possible~~

mapping of the data's descriptive statistics into its release parameters via a hierarchical series of learned affine mappings and nonlinearities.

The architecture of the MLP is defined by its input layer which receives the 5-dimensional descriptor vector, followed by 7 layers of feed forward hidden layers having widths of [256,512,1024,1024,512,256,128]. Each layer is followed by layer normalization and the Gelu activation function. Every 2 layers there are residual connections that allow gradients to pass through all layers without vanishing. After the last hidden layer, there is one additional linear layer with 7 outputs representing the 6 release parameters. For training, we use a mean-squared error loss function based on the 7 output values.

$$L_{MLP} = \left(\frac{1}{N} \right) \sum_n \sum_k w_k (\theta_k^n - \hat{\theta}_k^n)^2 \quad (15)$$

We have chosen weights $w = (1.0, 0.8, 0.5, 0.5, 0.3, 0.3)$ that emphasize the first 2 parameters which represent speed and spin rate as they are the most important ones among all 6 parameters. We train our model using AdamW optimizer (15) with initial learning rate equal to $3e-4$, cosine annealing schedule for learning rate reduction during training, weight decay set at $1e-4$, batch size of 4096 samples and train the model for 100 epochs.

~

Recurrent Neural Network

~

Lastly, the same 3-layer MLP-head as the CNN produced a 6-dimensional output. As the network used a bi-directional architecture it was able to incorporate both early-flight velocity establishing information and late-flight braking profile information into its predictions.

~

Conclusion

~

They are called: multilayer perceptron (MLP); convolutional neural network (CNN); recurrent neural networks using gated recurrent units (GRU-RNN). They were compared to see how they would each treat the trajectory of the baseball. The MLP treated the trajectory as a single vector; the CNN treated the trajectory like a spatial map where the characteristics of the trajectory could provide additional data about the magnitude of the Magnus force developed during its flight; the GRU-RNN treated the trajectory as a time sequence and provided the ability to infer a portion of the trajectory from other parts.

~

References

1. Nathan, A. M. The physics of baseball. Phys. Today 2003, 56 (10), 32–37.
2. Bahill, A. T.; Baldwin, D. G.; Venkateswaran, J. Predicting a baseball's path. Am. Sci. 2005, 93 (3), 218–225.
3. Tikhonov, A. N.; Arsenin, V. Y. Solutions of Ill-Posed Problems; Winston: Washington, DC, 1977.
4. Driveline Baseball. Modern pitch design: translating movement goals into mechanical instructions. Driveline Baseball Research Blog, 2022.
5. Seydel, R. Practical Bifurcation and Stability Analysis, 3rd ed.; Springer: New York, 2010.
6. Hadamard, J. Sur les problèmes aux dérivées partielles et leur signification physique. Bull. Univ. Princeton 1902, 13, 49–52.
7. Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning Representations by Back-Propagating Errors. Nature 1986, 323 (6088), 533–536.
8. LeCun, Y.; Bengio, Y.; Hinton, G., 2015. Deep learning. Nature, 521(7553), pp.436-444.
9. Elman, J. L. Finding Structure in Time. Cogn. Sci. 1990, 14 (2), 179–211.

10. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555. <https://arxiv.org/abs/1412.3555>.
11. Baseball Savant. Baseball Savant: Statcast, Trending MLB Players and Visualizations; Major League Baseball: New York, NY. <https://baseballsavant.mlb.com/> (accessed 2026-07-28).
12. Petti, J.; Gilani, S. pybaseball: a Python package for baseball data. *J. Open Source Softw.* **2021**, *6* (59), 3234.
13. Nathan, A. M. The Effect of Spin on the Flight of a Baseball. *Am. J. Phys.* **2008**, *76* (2), 119–124.
14. urden, R. L.; Faires, J. D.; Burden, A. M. Numerical Analysis, 11th ed.; Cengage Learning: Boston, MA, 2019.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *Proc. IEEE CVPR* **2016**, 770–778.
16. Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*; 2015.
17. dos Santos, C.; Tan, M.; Xiang, B.; Zhou, B. Attentive Pooling Networks. *arXiv* **2016**, arXiv:1602.03609. <https://arxiv.org/abs/1602.03609>.
18. Owens, N.; Harris, C.; Stennett, C. Hawk-Eye: A System for Tracking and Analyzing Ball Trajectories in Sports. In *Proceedings of the International Conference on Visual Information Engineering (VIE)*; IEEE: London, U.K., 2003; pp 638–642.

Reviewer #2

Comment #1

Please verify that all links to the references point to the correct source.

Response:

We thank the reviewer for this feedback. We acknowledge that there has been a lot of inconsistencies in the references. We verified all the citations in the manuscript to point to the correct source one by one.

Comment #2

The authors must disclose and acknowledge any assistance received in the preparation of this manuscript, including but not limited to editorial, technical, analytical, or writing support. All such contributions must be clearly stated in the Acknowledgments section.

Response:

We thank the reviewer for raising this concern. This work was primarily done by Kyu Beom Kim with supervision from Subok Lee. We did not receive any assistance in the form of editorial, technical support.

Comment #3

Include enough recent references along with foundational ones. Ensure references directly support your claims. Avoid “padding.” Use credible sources (peer-reviewed journals, reputable books, official reports).

Response:

We agree that our reference list should be improved. We updated our references to use credible sources and included foundational works to support our claims. Also, references that did not directly support our claims were removed. However, due to the simplicity of the neural networks we chose for our study, we were not able to include the most recent works.

Comment #4

All assumptions (including implicit assumptions) must be explicitly stated and clearly justified. The authors should explain why each assumption is reasonable and discuss its impact on the results and conclusions.

Response:

We thank the reviewer for raising this concern. There were cases where assumptions were not fully justified and we added additional context to make each assumption reasonable. We also included the impact of those assumptions on the analysis of our experiments.

Comment #5

Avoid overreaching conclusions that extend beyond what is supported by the data and analysis.

Response:

We agree that some of our statements and claims were overstated. We clarified our language to explicitly state what was achieved and what the reader should be wary about. In instances where the distinctions were not clear, we added additional discussions on those issues. Below are the changes we made to our manuscript.

Experiments

Dataset

~

For both our CNN and RNN models, we reconstructed the full 50-point trajectory sequence X by numerically integrating the equations of motion from the provided initial conditions forward in time. The Statcast-fitted constant accelerations were used to verify that the numerical integration produced accurate results. Therefore, we obtained a per-pitch trajectory sequence of size (50, 3) with sub centimeter level of accuracy compared to the measurement of the HawkEye system. **We note that this reconstruction procedure means the CNN and RNN are trained to invert trajectories generated by our own forward aerodynamic model rather than independently measured full-flight position data; discrepancies between this simplified model (e.g., its treatment of seam orientation, atmospheric conditions, and seam-shifted wake effects) and true baseball aerodynamics are consequently not captured during training, and could cause the sequence-based models to learn to invert the assumed physics rather than the full complexity of real flight.**

Comment #6

Verify that you have used past perfect tense and third person throughout the manuscript wherever applicable.

Response:

We agree that there are some instances where past perfect tense and third person was not used in the manuscript. We checked our manuscript for those instances and corrected them.

The manuscript has improved substantially in response to the previous review, and I appreciate the authors' considerable effort in revising both the scientific framing and the experimental evaluation. Several of my earlier concerns have been addressed appropriately. In particular, the authors have corrected the previous implication that the models recover how a pitcher physically throws the ball and now more accurately describe the predicted quantities as aerodynamic release parameters rather than pitcher biomechanics. They have also explicitly acknowledged that low forward-trajectory error does not establish recovery of the unique true physical release state, clarified that the CNN/RNN trajectories are reconstructed using the authors' forward aerodynamic model rather than

independently measured full-flight trajectories, added a nonlinear least-squares (Levenberg–Marquardt) inverse-optimization baseline, and expanded the ensemble uncertainty analysis. These changes represent meaningful improvements and make the manuscript considerably clearer and more scientifically defensible. For example, the revised discussion now correctly states that a low forward error demonstrates that the inferred solution produces a similar trajectory, rather than demonstrating recovery of the true physical release state.

Nevertheless, I believe that one of the most important issues raised in the previous review remains unresolved, and it now also bears directly on the scientific novelty of the manuscript. The authors acknowledge that the inverse mapping is non-unique, but they have elected not to experimentally characterize that non-uniqueness. My previous review specifically proposed sampling physically plausible release-parameter vectors, propagating these through the authors’ existing forward simulator, retaining those that reproduce a target trajectory within a specified tolerance, and examining the resulting admissible solution space. The authors agree that such an analysis is feasible and would strengthen the manuscript, but state that it would increase computational effort and therefore defer it to future work. I do not consider acknowledgment of the suggestion and deferral to future work equivalent to addressing this concern.

This issue is particularly important because, without such an analysis, the manuscript is principally demonstrating that the baseball trajectory problem can be defined and learned in the inverse direction. The forward problem maps a release state to a resulting trajectory; the authors train supervised machine-learning models to approximate the reverse relationship, mapping trajectory information to a release-parameter vector. Comparing an MLP, CNN, GRU, ensemble, linear regression, gradient boosting, and nonlinear least-squares inversion is useful from a computational perspective, and the reported differences in performance are informative. However, demonstrating that a sufficiently flexible supervised model can learn an approximate inverse mapping from paired trajectory and release-parameter data is not, by itself, a particularly strong scientific discovery. Table 1 primarily establishes comparative predictive performance, with the ensemble achieving the lowest reported parameter errors and low forward-trajectory errors.

What potentially makes this problem scientifically interesting is precisely the feature that the authors currently defer to future work: the inverse may not be unique. The scientifically important question is therefore not simply whether a model can return a release state corresponding to a desired trajectory, but rather what family of release states can produce essentially the same trajectory, which release parameters are identifiable, which are mutually compensatory, and whether this degeneracy differs systematically among pitch types. Characterizing that structure would transform the manuscript from primarily a comparison of inverse regression algorithms into an investigation of the underlying inverse physics of baseball trajectories.

For example, a particular trajectory might constrain release velocity very tightly while permitting substantial compensatory variation between spin rate and spin efficiency. Another trajectory might permit correlated changes in release angle and spin axis. Some pitch types might possess a narrow, approximately point-like feasible region, whereas others might possess an elongated manifold of admissible solutions or even several distinct regions of parameter space. Demonstrating these possibilities empirically would reveal information that cannot be obtained simply by reporting the parameter vector selected by a trained neural network. More importantly, differences in the geometry or dimensionality of these admissible regions among four-seam fastballs, curveballs, changeups, splitters, sweepers, and other pitch types could constitute a genuinely novel scientific result.

This is also why I am not persuaded by the authors’ argument that characterization of the admissible solution cloud is inconsistent with the stated objective of producing an easy-to-use or computationally rapid inversion method. The proposed degeneracy experiment does not need to become part of the operational inference pipeline. The neural network can continue to provide a rapid single prediction during deployment. The admissible-solution analysis can instead be performed offline on representative test trajectories for purposes of scientifically characterizing the inverse problem. Consequently, the computational expense of generating a feasible-solution cloud

does not appear to be a compelling reason for omitting the experiment from the scientific validation of the method.

The absence of this experiment also creates a significant inconsistency between the actual experiments and the manuscript's present claims. The revised Introduction states that the authors "investigated the ill-posedness structure of the inverse map" and found regions where disagreement among architectures indicates "actual solution nonuniqueness." I do not believe the present analysis supports this statement. Disagreement among an MLP, CNN, and GRU does not demonstrate physical non-uniqueness of the inverse problem. These architectures have different inductive biases, receive different representations of the input, and may have different approximation and generalization errors. Consequently, they can disagree even if the underlying physical inverse solution were tightly constrained. Conversely, several models could converge on similar predictions because they learned similar statistical priors from the same training population even when multiple physically admissible solutions actually exist.

This distinction should also be made explicit in interpreting the newly added ensemble uncertainty analysis. The manuscript defines calibration using the Spearman correlation between ensemble standard deviation and actual prediction error. The authors find, for example, relatively high calibration correlations for four-seam fastballs and curveballs and lower values for splitters and changeups. This is an interesting result, but it measures whether model disagreement predicts model error. It does not establish that model disagreement measures physical inverse degeneracy. These are distinct concepts and should not be conflated.

Indeed, the proposed degeneracy analysis could provide a particularly valuable opportunity to test this hypothesis directly. For representative pitches, the authors could calculate both the ensemble disagreement and the width or volume of the admissible physical solution space. If pitches with high ensemble disagreement also possess broad feasible parameter regions, the authors would have evidence that ensemble uncertainty provides a practical surrogate for inverse identifiability. Conversely, if the two quantities are poorly associated, that result would itself be scientifically informative because it would demonstrate that neural-network uncertainty and physical non-identifiability should not be interpreted interchangeably. Either result would substantially strengthen the paper.

I therefore continue to recommend an empirical characterization of the admissible release-parameter space. This need not require an exhaustive analysis of the entire dataset. A representative analysis involving several qualitatively different pitch types would be sufficient to establish the principle. The authors already possess the essential components required for this experiment: a forward aerodynamic simulator, realistic parameter ranges, target trajectories, and quantitative trajectory-error measures. For each representative target, physically plausible parameter vectors could be sampled globally or locally around the predicted solution, propagated through the forward model, and retained when the reconstructed trajectory falls within a predefined tolerance of the target. The resulting accepted parameter vectors could then be visualized using pairwise parameter plots and summarized using covariance analysis, principal-component analysis, or another suitable dimensionality-reduction method. This would reveal which parameter combinations are tightly constrained and which exhibit compensatory relationships.

The authors should also specify the trajectory tolerance used to define an admissible solution and examine whether the inferred degeneracy is robust to reasonable changes in this threshold. This is important because an inverse solution space is necessarily conditional on what constitutes an acceptably equivalent trajectory. Showing the solution cloud at more than one tolerance would provide a much more meaningful picture of identifiability than simply stating mathematically that the forward mapping is non-injective.

The distinction between the scalar MLP inversion and the sequence-based CNN/RNN inversion also deserves particular attention in this context. The manuscript notes that the scalar forward map has six release parameters mapped onto five trajectory descriptors and therefore argues that the mapping is non-injective, whereas the sequence models receive a much richer 50×3 representation. Consequently, the degree of practical non-identifiability need not be the same for these two

representations. It would be scientifically interesting to determine whether access to the full trajectory collapses some of the degeneracy present when only scalar trajectory descriptors are available. This could provide a much stronger mechanistic explanation for why the CNN and GRU perform differently from the MLP than simply comparing their prediction errors.

The newly added nonlinear least-squares analysis is a welcome addition and directly responds to my previous request for comparison with a conventional inverse-optimization method. The reported NLS solution performs substantially worse than the neural-network architectures, with release-speed, spin-rate, and spin-axis MAEs of 2.89 mph, 267 RPM, and 6.5°, respectively, together with much larger forward IVB and HB errors. However, the authors should moderate the interpretation of this result. Performance of nonlinear least-squares in an ill-conditioned inverse problem may depend strongly on initialization, parameter scaling, imposed physical bounds, convergence criteria, regularization, and whether multistart optimization is used. The manuscript should therefore provide sufficient methodological information about the NLS implementation to make the comparison reproducible. The result establishes that the tested neural networks outperform the specific NLS implementation evaluated here; it should not be generalized into a broader conclusion that neural-network inversion is inherently superior to classical inverse optimization.

There is also a conceptual issue with treating deviation from the recorded Statcast release vector as straightforward “parameter error” while simultaneously emphasizing that the inverse problem is non-unique. If two substantially different release vectors produce trajectories that are indistinguishable within the chosen physical tolerance, then one of those solutions should not necessarily be considered incorrect merely because it differs from the recorded release state. The recorded Statcast parameters provide an appropriate reference for supervised prediction, but they are not necessarily the unique solution of the inverse problem. The manuscript should therefore distinguish recovery of the observed/reference release state from physical validity of an inverse solution. Again, experimentally characterizing the admissible parameter space would resolve much of this conceptual ambiguity.

The authors have adequately addressed my previous concern regarding the reconstructed full trajectories. The revised manuscript now explicitly acknowledges that the CNN and RNN are trained to invert trajectories generated by the authors’ own forward aerodynamic model rather than independently measured full-flight positional data, and appropriately notes that effects such as seam orientation, atmospheric conditions, and seam-shifted wakes may not be represented. This limitation should remain prominent because excellent inversion of trajectories generated using the same underlying physics model is not equivalent to demonstrating equivalent performance on trajectories containing aerodynamic phenomena absent from that model.

There are also unresolved editorial and methodological inconsistencies that must be corrected. Most importantly, the response letter states that the duplicated MLP material and the previous inconsistency regarding six versus seven output parameters were corrected. However, both problems remain in the revised manuscript. The MLP architecture is described twice in consecutive passages, and the manuscript still states that the output layer contains “7 linear neurons corresponding to the six release parameters” and subsequently refers to “7 outputs representing the 6 release parameters.” This is particularly problematic because the release vector is explicitly formulated as six-dimensional elsewhere in the manuscript, and the loss weights contain six values. The authors must establish whether six or seven quantities are actually predicted and make the architecture description, equations, loss function, tables, and implementation mutually consistent. The duplicated methodological description should also be removed. These are not merely cosmetic concerns because the dimensionality of the predicted inverse state is fundamental to the mathematical formulation of the problem.

More generally, the manuscript would benefit from another careful editorial review. Several passages remain awkward or overly categorical, and terminology concerning “prediction,” “recovery,” “inversion,” “uncertainty,” “identifiability,” and “non-uniqueness” should be used consistently. In particular, “recovering the release parameters” can imply identification of the true generating state. Where uniqueness has not been established, language such as “inferring a plausible

release state” or “predicting a release-parameter vector consistent with the target trajectory” would be scientifically more precise.

My recommendation therefore remains Major Revision. The manuscript is substantially improved and contains a potentially interesting study, but the revision has not yet resolved the question that most strongly determines its scientific novelty. If the authors experimentally characterize the admissible inverse solution space—even for a representative subset of pitch types—the study could move beyond simply implementing and comparing algorithms for a reversed trajectory-to-parameter mapping and instead provide new insight into the structure, identifiability, and parameter degeneracies of the baseball trajectory inverse problem. The particularly interesting question would then become not merely which neural network predicts the recorded parameters most accurately, but which physical properties of a pitch can actually be inferred uniquely from its trajectory and which cannot.

If the authors elect not to perform this experiment, the manuscript can still report a useful computational result, but its novelty and claims must be correspondingly reduced. In that case, it should be reframed explicitly as a comparative study of machine-learning architectures for rapid approximate inverse prediction. Claims that the work experimentally characterizes the “ill-posedness structure,” demonstrates “actual solution nonuniqueness,” or resolves the baseball inverse problem should be removed or substantially weakened. The distinction is important: defining and learning the problem in the inverse direction is not equivalent to scientifically characterizing the inverse problem itself.

Reviewer #1

We want to express our gratitude to the reviewer for the thoughtful and useful second round of comments. The reviewer made an important point about the lack of experimental characterization of the non-uniqueness of the inverse mapping. After some consideration, we agreed with the reviewer, and found the deferral to future research not acceptable. Thus we carried out the “admissible solution space” experiment suggested by the reviewer in his original comments and included the results as a new subsection entitled “Admissible Solution Space Analysis” along with two new figures (Figures 2 and 3). In addition, we modified how the paper's claims were framed based upon what this new data establishes or does not establish. Furthermore, we corrected the two remaining editorial problems that had previously been mistakenly identified as solved; specifically, the duplicate description of the Multi-Layer Perceptron model and the inconsistency regarding whether there are six outputs or seven.

Below we address each comment individually.

Comment #1: non-uniqueness was not experimentally characterized

“The authors agree that such an analysis is feasible and would strengthen the manuscript, but state that it would increase computational effort and therefore defer it to future work. I do not consider acknowledgment of the suggestion and deferral to future work equivalent to addressing this concern.”

Response:

We agree and have run analysis of this issue. We used the same forward aero model (equations 1-11) that was used to generate our training targets for sequence-model training and sampled 10,000 physically plausible release-parameter vectors around each representative release state for each pitch type (four-seam fastball, curve-ball, sweeper, splitter). We then propagated these vectors through the forward model and retained those whose resulting trajectory matched within tolerance. These admissible clouds are visualized and analyzed in new “Admissible Solution Space Analysis” subsection (new Figures 2-3) inserted immediately after the “pitch type comparison” section. This work was done offline outside of deployment inference pipeline as you suggested.

Comment #2: Introduction overclaimed what the manuscript's experiments showed

"The revised Introduction states that the authors "investigated the ill-posedness structure of the inverse map" and found regions where disagreement among architectures indicates "actual solution nonuniqueness." I do not believe the present analysis supports this statement."

Response:

We both agreed to delete the sentence which wasn't supported by the available data as of then. The Introduction has been revised to describe only what we really did. The Introduction states: "We also used a similar aerodynamic forward model that produces our training target and sampled the release parameter space for each of four examples of different pitches. We did these samples at multiple levels of tolerance for trajectory matching and at two different conditions regarding how much information was provided." Both the Abstract and the Conclusion were changed accordingly with appropriate wording for matching.

Comment #3: Model disagreement is not physical non-uniqueness

"Disagreement among an MLP, CNN, and GRU does not demonstrate physical non-uniqueness of the inverse problem... they can disagree even if the underlying physical inverse solution were tightly constrained."

Response:

In agreement with this conceptualization, we have made no use of "inter-architecture" disagreements as evidence of nonuniqueness throughout the revised manuscript. Rather, our new admissible solution space analysis provides a direct measurement of nonuniqueness based solely on the physics contained within the forward model; it does so independently of the behavior of any given trained networks. When comparing our physical measure of nonuniqueness to the calibration of an ensemble (point 4 below) we explicitly identify this comparison as one being between two fundamentally different quantities, rather than claiming that these quantities are equivalent.

Comment #4: Ensemble calibration measures disagreement-predicts-error, not physical degeneracy

"This is an interesting result, but it measures whether model disagreement predicts model error. It does not establish that model disagreement measures physical inverse degeneracy. These are distinct concepts and should not be conflated... Indeed, the proposed degeneracy analysis could provide a particularly valuable opportunity to test this hypothesis directly."

Response:

We actually ran the exact experiment that was suggested by the reviewer. The relative sizes of the regions for which we could find an acceptable fit to the motion data using the scalar descriptor criterion were: splitter > sweeper > curveball $\approx$ four-seam fastball at each value of tolerance examined. This is essentially the inverse of our previous table (Table 2) showing the ensemble calibration values ($\rho=0.74, 0.82, 0.92, 0.93$ respectively), and we have included this information in a new section of the paper as preliminary support at the level of individual pitches, that both ensemble differences in predictions of physical properties and physical identifiability are related to one another. We also specifically state that since this analysis only includes data from four representative pitches it should not be taken as a generalization to all pitches.

Comment #5: Tolerance dependence of the admissible solution space

"The authors should also specify the trajectory tolerance used to define an admissible solution and examine whether the inferred degeneracy is robust to reasonable changes in this threshold."

Response:

The new study includes a number of results at four different levels of allowable error per acceptable criteria ($\epsilon = 0.25, 0.5, 1.0, 2.0$ in for the scalar criteria; $\epsilon = 1.5, 2.5, 4.0, 6.0$ in for the entire trajectory criteria) and also has a new figure two showing the area of admissible regions as a function of allowable error for all four pitch types. We found that the ordering of pitch types based on admissible region size was consistent over this range, and we explicitly noted that while the size of the sampling box affects the actual admissible fraction values, it does not appear to affect the order in which these fractions rank.

Comment #6: Scalar vs. full-trajectory identifiability

"It would be scientifically interesting to determine whether access to the full trajectory collapses some of the degeneracy present when only scalar trajectory descriptors are available."

Response:

This is the third main finding of this new subsection. In terms of the scalar descriptor criterion, the release angle parameters (ϕ_{rel}, θ_{rel}) were relatively un-constrained due to the sampled box at all tolerances evaluated. For the full-trajectory criterion, the same two parameters were constrained approximately 4-6 times tighter relative to an admitted fraction, of equal magnitude. We present this as mechanistic explanation for why the CNN & RNN outperformed the MLP in Table 1. In particular, we assert that the richer input representation specifically disambiguated release direction rather than simply containing more information in general sense. As negative result of comparable interest, we report that release speed, spin rate & spin efficiency remain broadly degenerate with respect to both criteria, which we attribute directly to the $|v|^2 \cdot c_l(s)$ product structure of the Magnus force term (equation 4). This is visualized through a consistent negative correlation between spin rate and spin efficiency within the sampled admissible clouds (Pearson $r = -0.44$ to -0.74 across the four pitch types) in new figure 3

Comment #7: NLS baseline needs implementation detail and a scoped conclusion

"Performance of nonlinear least-squares in an ill-conditioned inverse problem may depend strongly on initialization, parameter scaling, imposed physical bounds, convergence criteria, regularization, and whether multistart optimization is used... it should not be generalized into a broader conclusion that neural-network inversion is inherently superior to classical inverse optimization."

Response:

We added the implementation details of the NLS (Nonlinear Least Squares) analysis to include that the initial conditions are set as the population mean of the release vectors used to generate the target pitch types. In addition we included that the parameters were scaled to be within the range $[0,1]$. The convergence criteria is now set to $1e-8$; the number of restarts has been reduced to zero. Also the scope of the last sentence was changed by adding "for this configuration" so it refers only to this particular model and not to all inverse optimizations.

Comment #8: Statcast reference vs. physical validity of a solution

"If two substantially different release vectors produce trajectories that are indistinguishable within the chosen physical tolerance, then one of those solutions should not necessarily be considered incorrect merely because it differs from the recorded release state."

Response:

We have also included a new paragraph near where we discuss the NLS to clarify these differences in terminology; namely, the record Statcast release vector is a suitable supervised learning target and practical reference point, however, it does not represent a unique solution to the inverse problem; therefore, if you make a prediction that does not match this target vector yet still produces the same target trajectory (within some tolerance) your prediction is not wrong.

Comment #9: Duplicated MLP description and six-vs-seven output inconsistency

"The MLP architecture is described twice in consecutive passages, and the manuscript still states that the output layer contains "7 linear neurons corresponding to the six release parameters" and subsequently refers to "7 outputs representing the 6 release parameters."

Response:

The duplicate paragraphs and equations have been completely removed. The six linear units of the output layer are now described consistently with a total of six units that match both the six-dimensional release parameter vector from Equation (12), and the six loss weights in Equation (15).

Comment #10: Terminology precision

"In particular, "recovering the release parameters" can imply identification of the true generating state... language such as "inferring a plausible release state" or "predicting a release-parameter vector consistent with the target trajectory" would be scientifically more precise."

Response:

We looked at every occurrence of "recover"/"recovers" within the document. Only one occurred; it was located in the caption for Fig. 1. In this caption, we had written that "the inverse problem recovers release parameters from a desired trajectory" and have since revised it to "the inverse problem infers a plausible release-parameter vector consistent with a desired trajectory". There is another location where we were correct on our wording, as it was phrased "does not necessarily imply the recovery of the true physical release state", and therefore we did nothing further.

Thank you for addressing my comments. **Accepted** with the following three caveats:

1. the spin efficiency equation is corrected
2. the following paragraph be added to the "limitations section" :

A further limitation concerns the comparison between the scalar and full-trajectory admissible solution spaces. The scalar admissibility analysis was defined using IVB and HB, whereas the MLP receives five trajectory descriptors (IVB, HB, plate velocity, and horizontal and vertical plate location). Consequently, the broader admissible regions obtained under the IVB/HB criterion cannot be interpreted as a direct information-matched explanation for the performance difference between the MLP and sequence-based CNN/RNN models. The additional MLP inputs may themselves constrain release velocity, release direction, or other components of the admissible parameter space. The present analysis therefore demonstrates that full-trajectory information can reduce degeneracy relative to IVB/HB alone, but does not establish that this reduction accounts for the observed CNN/RNN advantage over the MLP. Future work should construct an admissible-space criterion incorporating all five MLP inputs and compare it directly with the full-trajectory criterion.

and

3. A "Perspectives and future applications section" be included along the lines below:

Perspectives and Future Application

The non-uniqueness of the inverse problem may have practical implications beyond limiting the recovery of a unique release state. If multiple combinations of release speed, spin rate, spin efficiency, and release direction can produce essentially the same trajectory, the admissible solution space can instead be regarded as a pitch-design space. Importantly, the present results suggest that the extent of this flexibility may differ among pitch types. Among the representative pitches examined here, the splitter exhibited a substantially larger admissible release-parameter region than the sweeper, curveball, and four-seam fastball, suggesting that different pitch types may impose different degrees of constraint on how their characteristic trajectories can be produced. A pitch with a broad admissible region may permit substantial compensation among release parameters, whereas a more tightly constrained pitch may require a narrower combination of release conditions. This distinction could have direct implications for coaching and individualized pitch development.

Rather than prescribing an idealized release condition, a coach could potentially identify the range of release states capable of producing the desired pitch and select the combination most compatible with an individual pitcher's velocity, spin-generation ability, spin efficiency, release mechanics, and reproducibility. The same framework could eventually provide an additional dimension for player evaluation and valuation. Expertise in a tightly constrained pitch may represent a different technical capability from proficiency in a pitch possessing a broad admissible solution space, while a pitcher capable of reliably producing several pitch types with differently constrained release requirements may possess greater adaptable pitching capacity. Conversely, a player's apparent inability to reproduce a particular pitch need not imply a general limitation if alternative admissible release strategies exist that better match that player's capabilities. Thus, inverse-space characterization could ultimately complement conventional outcome-based statistics by distinguishing not only what pitches a player can produce, but also the physical constraints, flexibility, and reproducibility underlying that repertoire. These coaching and player-evaluation applications remain hypotheses rather than demonstrated consequences of the present study. Future work should therefore establish whether pitch-type-specific admissible spaces persist across larger populations, determine whether pitchers can deliberately exploit predicted compensatory release strategies while preserving target trajectories, and test whether the size and structure of these spaces are associated with pitch learnability, reproducibility, performance, or player development outcomes.