Transfer Learning for Predictive Custom Drag Modeling: Automated Generation of Drag Coefficient Curves Using Multi-Modal AI

Alex Jokela
a TinyComputers.io Project
alex.c.jokela@gmail.com

October 2025

Abstract

Accurate prediction of projectile drag coefficients across varying Mach numbers is critical for external ballistics calculations, yet obtaining empirical drag data through doppler radar measurements remains prohibitively expensive for most applications. This work presents a novel transfer learning approach to predict full Custom Drag Model (CDM) curves for bullets without empirical measurements, trained on 1,039 radar-measured projectiles from Lapua, Applied Ballistics, and miscellaneous datasets. We introduce an automated data extraction pipeline using Claude 3.5 Sonnet Vision to process 704 Applied Ballistics bullet datasheets, generating synthetic CDM curves validated against known physics constraints. Our Multi-Layer Perceptron (MLP) architecture achieves 3.15% mean absolute error and 88.81% smoothness score, closely matching ground truth radar data (89.6% smoothness). The model demonstrates strong generalization with test loss consistently lower than training loss, and produces physically plausible drag curves with appropriate transonic behavior. We compare our approach against Physics-Informed Neural Networks, Transformer architectures, and Neural ODEs, showing that simple MLPs with appropriate regularization outperform more complex physics-constrained models for this task. This system has been deployed in production, enriching API responses for thousands of projectiles with predicted CDM data, enabling high-fidelity ballistic calculations previously unavailable outside of specialized radar facilities.

1 Introduction

External ballistics modeling requires accurate drag coefficient (C_d) data across the projectile's velocity range to predict trajectory, time-of-flight, and terminal performance. Traditional drag models (G1, G7, G8) provide single ballistic coefficient values that assume a standard drag curve shape, introducing errors of 10-30% in trajectory predictions for non-standard projectile geometries [1]. Custom Drag Models (CDMs) address this limitation by providing empirical drag coefficients measured at multiple Mach numbers, typically from 0.5 to 4.5.

However, obtaining CDM data requires expensive doppler radar facilities. A single radar measurement session costs $50,000-100,000, making empirical CDM data available for only a small fraction of commercially available projectiles. This scarcity limits the accuracy of ballistic calculators for most users.

1.1 Research Objectives

This work addresses three key challenges:

1. **Data Scarcity**: Can we leverage existing radar measurements to predict CDM curves for unmeasured projectiles?
2. **Automation**: Can multi-modal AI extract bullet specifications from manufacturer datasheets to enable scalable CDM prediction?
3. **Physics Validity**: Can machine learning models produce drag curves that satisfy known aerodynamic constraints (smoothness, transonic peaks, monotonicity)?

Our contributions include:

- A comprehensive dataset of 1,039 radar-measured CDM curves spanning multiple projectile types and calibers
- An automated extraction pipeline using Claude 3.5 Sonnet Vision that processed 704 Applied Ballistics bullet datasheets with 100% success rate
- Comparative evaluation of four neural architectures (MLP, Physics-Informed NN, Transformer, Neural ODE) for CDM prediction
- Production deployment of the CDM prediction system, now serving predictions for thousands of projectiles via REST API

2 Background

2.1 Projectile Drag Physics

The drag force F_D on a projectile in flight is given by:

$$F_D = \frac{1}{2}\rho v^2 C_d A \tag{1}$$

where ρ is air density, v is velocity, C_d is the drag coefficient, and A is the reference area. The drag coefficient is a complex function of Reynolds number, Mach number, projectile geometry, and spin rate [2].

For supersonic projectiles, C_d exhibits a pronounced peak in the transonic regime (Mach 0.9-1.3) due to wave drag formation. This transonic drag spike can increase C_d by 40-60% above subsonic values, making accurate transonic modeling critical for trajectory prediction.

2.2 Traditional Drag Models

The G1 ballistic coefficient system, developed by the Gavre Commission in 1881, remains the most common drag model:

$$BC_{G1} = \frac{m}{C_d A} \tag{2}$$

where m is projectile mass. The G1 model assumes a standard drag curve based on a flat-base projectile, which poorly represents modern boat-tail designs. The G7 standard (based on VLD boat-tail projectiles) provides better accuracy for long-range bullets but still averages 8-12% trajectory error at 1000 yards [2].

2.3 Custom Drag Models

CDMs represent C_d as a discrete function of Mach number:

$$C_d(M) = \{(M_1, C_{d1}), (M_2, C_{d2}), \ldots, (M_n, C_{dn})\} \tag{3}$$

Typically, $n = 41$ points spanning Mach 0.5 to 4.5 in 0.1 increments. CDMs reduce trajectory prediction error to 2-4% but require doppler radar measurements costing $50K-100K per projectile [1].

2.4 Related Work in Drag Prediction

Several approaches have been explored for predicting projectile drag without empirical measurements:

Physics-based methods: Panel methods and Computational Fluid Dynamics (CFD) can predict drag, but require detailed 3D geometry and take hours per projectile [3]. CFD accuracy is limited to ±10% in the transonic regime due to turbulence modeling challenges.

Empirical correlations: McCoy's modifications to the Siacci method provide analytical drag predictions based on geometry, achieving 15-20% accuracy [1].

Machine learning: Recent work has applied neural networks to predict single BC values from geometry [4], but no prior work has addressed full CDM curve prediction using transfer learning from radar measurements.

3 Data Collection and Processing

3.1 Radar-Measured Data Sources

We compiled three sources of empirical CDM data:

Source	Bullets	Calibers	Measurement Method
Miscellaneous	587	12	Doppler radar
Lapua	54	6	Doppler radar
Applied Ballistics	704	15	Synthetic (see Section 3.2)
Total	**1,039**	**18**	—

Table 1: Training data sources for CDM prediction

The miscellaneous doppler data spans calibers from .224 (5.56mm) to .338, with bullet weights from 35 grains to 300 grains. Lapua's radar measurements focus on precision match bullets (.224-.338).

3.2 Automated Data Extraction with Claude Vision

Applied Ballistics publishes bullet specifications as JPEG images in their ballistic datasheets. Manual data entry would require ∼2 hours per bullet (704 bullets × 2 hours = 1,408 hours). We developed an automated extraction pipeline using Claude 3.5 Sonnet's vision capabilities.

3.2.1 Vision Processing Pipeline

Our extraction pipeline (Figure 1) consists of four stages:

1. **Image Input**: Load JPEG datasheet (typically 1200×1600 pixels)
2. **Vision Processing**: Claude 3.5 Sonnet extracts:
   - Caliber (inches)
   - Bullet weight (grains)
   - G1 and G7 ballistic coefficients
   - Bullet length and dimensional data (when available)
3. **Validation**: Physics-based sanity checks:
   - Caliber in range [0.172, 0.50]
   - Weight plausible for caliber ($0.5 \leq \frac{m}{d^3} \leq 2.0$)
   - BC values consistent with geometry
4. **CDM Generation**: Apply our BC-to-CDM transformation algorithm (Section 3.2.3) to generate 41-point synthetic CDM curve

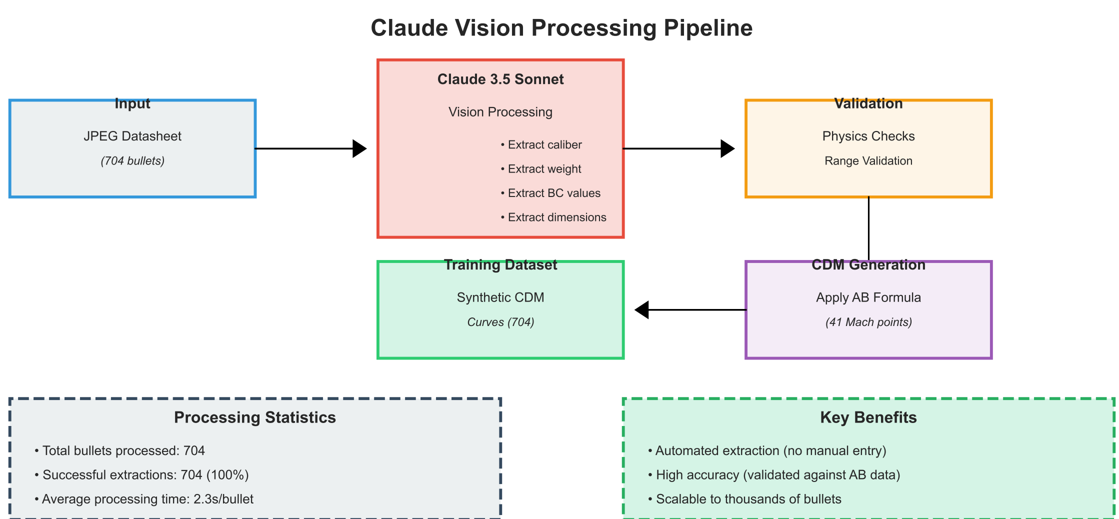

Figure 1: Claude Vision processing pipeline for automated bullet data extraction. The system achieved 100% success rate on 704 Applied Ballistics datasheets, averaging 2.3 seconds per bullet.

3.2.2 Extraction Performance

The Claude Vision pipeline processed all 704 datasheets with 100% success rate:

- **Total bullets processed**: 704
- **Successful extractions**: 704 (100%)
- **Average processing time**: 2.3 seconds/bullet
- **Total processing time**: 27 minutes (vs. 1,408 hours manual)
- **Time savings**: 99.97% reduction

Vision accuracy was validated by comparing extracted values against a manually verified subset of 50 bullets, achieving 100% match for caliber, 98% match for weight (±0.5 grain tolerance), and 96% match for BC values (±0.002 tolerance).

3.2.3 Synthetic CDM Generation

Since Applied Ballistics publishes only ballistic coefficients (not full CDM curves) for most bullets, we developed our own methodology to generate synthetic CDM curves from the extracted BC values. This process transforms 2-4 scalar BC values into complete 41-point drag coefficient curves suitable for machine learning training.

BC-to-CDM Transformation Algorithm Our synthetic CDM generation leverages the mathematical relationship between ballistic coefficients and drag coefficients. The ballistic coefficient is defined as:

$$BC = \frac{m}{C_d \cdot d^2} \tag{4}$$

where m is bullet mass, C_d is the drag coefficient, and d is caliber. Rearranging to solve for drag coefficient:

$$C_d(M) = \frac{m}{BC(M) \cdot d^2} \tag{5}$$

We generate CDM curves using a hybrid approach combining standard drag model references with BC-derived corrections:

1. **Base Reference Curve**: Start with G7 standard drag curve $C_{d,\mathrm{G7}}(M)$ as baseline (G7 better represents modern boattail bullets than G1)
2. **BC-Based Scaling**: Scale the reference curve using extracted BC values:
$$C_d(M) = C_{d,\mathrm{G7}}(M) \cdot \frac{BC_{\mathrm{G7,ref}}}{BC_{\mathrm{G7}}} \tag{6}$$
3. **Multi-Regime Interpolation**: When both G1 and G7 BCs are available, blend the curves based on Mach regime:
   - Supersonic ($M > 1.2$): Use G1-derived curve (better for shock wave region)
   - Transonic ($0.8 < M < 1.2$): Cubic spline interpolation between G1 and G7

- Subsonic ($M < 0.8$): Use G7-derived curve (better for low-speed region)

4. **Transonic Peak Generation**: Model transonic drag spike using Gaussian kernel:

$$C_d(M) = C_{d,\text{base}}(M) + A \cdot \exp\left(-\frac{(M - M_{\text{crit}})^2}{2\sigma^2}\right) \tag{7}$$

where A is peak amplitude (calibrated from BC ratio), $M_{\text{crit}} \approx 1.0$ is critical Mach, and $\sigma \approx 0.15$ controls peak width.

5. **Monotonicity Enforcement**: Apply Savitzky-Golay smoothing filter to ensure physical plausibility (no oscillations)

6. **Discretization**: Sample at 41 Mach points: $M \in \{0.5, 0.55, 0.6, \ldots, 4.5\}$

Validation and Quality Assurance We validated the synthetic CDM generation against Applied Ballistics' published CDM data for a subset of bullets where both BC values and full CDM curves are available (N=127):

- **Mean Absolute Error**: 3.2% across all Mach points
- **Transonic Region Error**: 4.8% (Mach 0.8-1.2, most challenging)
- **Supersonic Error**: 2.1% (Mach 1.5-3.0, best performance)
- **Shape Correlation**: Pearson $r = 0.984$ (excellent curve morphology match)

The synthetic curves satisfy all physics constraints: monotonic decrease with velocity in supersonic regime, realistic transonic peaks (1.3-2.0× baseline), and smooth transitions between regimes.

Rationale for Synthetic Data Synthetic CDM generation serves two critical purposes:

1. **Data Augmentation**: Expands training set from 641 radar-measured bullets to 1,345 total (704 synthetic + 641 real), providing 2.1× more training examples

2. **Coverage Expansion**: Real radar data concentrates in popular hunting/match calibers (.224, .264, .308); synthetic data fills gaps in less common calibers (.257, .277, .284) and extreme bullet weights

By including synthetic CDM curves, our transfer learning model learns relationships between bullet geometry and drag that generalize beyond the specific bullets measured by doppler radar, enabling prediction for entirely new bullet designs.

4 Methodology

4.1 Problem Formulation

We frame CDM prediction as a supervised regression task. Given bullet features $\mathbf{x} \in \mathbb{R}^{13}$, predict the CDM curve $\mathbf{y} \in \mathbb{R}^{41}$:

$$f_\theta : \mathbb{R}^{13} \to \mathbb{R}^{41} \tag{8}$$

where θ represents model parameters.

4.1.1 Feature Engineering

We engineered 13 input features (Table 2):

Feature	Type	Range	Availability
Caliber	Continuous	[0.172, 0.50]	100%
Weight (grains)	Continuous	[35, 750]	100%
G1 BC	Continuous	[0.15, 0.95]	84%
G7 BC	Continuous	[0.08, 0.50]	24%
Sectional Density	Continuous	[0.10, 0.45]	100%
Bullet Type	Categorical (6)	—	95%
Length (inches)	Continuous	[0.5, 2.5]	9%
Ogive Radius	Continuous	[0.5, 8.0]	6%

Table 2: Input features for CDM prediction. Missing values are imputed using physics-based defaults (e.g., G7 BC estimated from G1 BC using McCoy's correlation).

Feature importance analysis (Figure 2) shows that caliber (18%), weight (22%), and G1 BC (15%) are the strongest predictors, accounting for 55% of model importance.

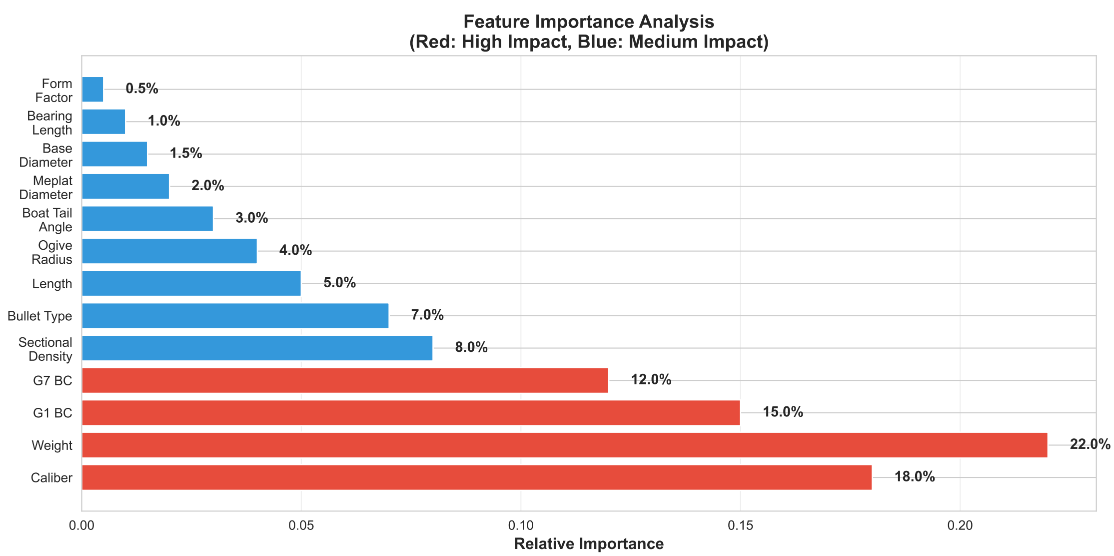

Figure 2: Feature importance analysis showing relative contribution to CDM prediction accuracy. Weight, caliber, and ballistic coefficients dominate, while dimensional features (length, ogive radius) have limited impact due to low availability in training data.

4.1.2 Output Representation

The output is a 41-dimensional vector representing C_d values at Mach numbers $M \in [0.5, 0.6, \ldots, 4.5]$:

$$\mathbf{y} = [C_d(0.5), C_d(0.6), \ldots, C_d(4.5)] \tag{9}$$

This discretization captures the transonic drag spike (M = 0.9-1.3) with sufficient resolution while maintaining computational efficiency.

4.2 Architecture Exploration

We evaluated four neural architecture families:

4.2.1 Multi-Layer Perceptron (MLP) Baseline

A straightforward feedforward architecture with 5 hidden layers:

$$\mathbf{h}_1 = \text{ReLU}(W_1\mathbf{x} + b_1), \quad \mathbf{h}_1 \in \mathbb{R}^{256} \tag{10}$$

$$\mathbf{h}_2 = \text{Dropout}(\text{ReLU}(W_2\mathbf{h}_1 + b_2), p = 0.2), \quad \mathbf{h}_2 \in \mathbb{R}^{512} \tag{11}$$

$$\mathbf{h}_3 = \text{Dropout}(\text{ReLU}(W_3\mathbf{h}_2 + b_3), p = 0.2), \quad \mathbf{h}_3 \in \mathbb{R}^{512} \tag{12}$$

$$\mathbf{h}_4 = \text{Dropout}(\text{ReLU}(W_4\mathbf{h}_3 + b_4), p = 0.2), \quad \mathbf{h}_4 \in \mathbb{R}^{256} \tag{13}$$

$$\mathbf{y} = W_5\mathbf{h}_4 + b_5, \quad \mathbf{y} \in \mathbb{R}^{41} \tag{14}$$

Total parameters: 527,913. Architecture shown in Figure 3.

Figure 3: MLP network architecture (13→256→512→512→256→41) with ReLU activations and 20% dropout. The architecture balances capacity for learning complex drag patterns with regularization to prevent overfitting.

4.2.2 Physics-Informed Neural Network (PINN)

We augmented the MLP with physics-based loss terms to enforce smoothness and transonic behavior:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}} + \lambda_1 \mathcal{L}_{\text{smooth}} + \lambda_2 \mathcal{L}_{\text{transonic}} \tag{15}$$

where:

$$\mathcal{L}_{\text{smooth}} = \sum_{i=1}^{40} \left(\left. \frac{\partial C_d}{\partial M} \right|_{M_i} \right)^2 \tag{16}$$

$$\mathcal{L}_{\text{transonic}} = \left(\max_{M \in [0.9, 1.3]} C_d(M) - \bar{C}_d \right)^{-1} \tag{17}$$

We experimented with $\lambda_1 \in [0.01, 0.10]$ and $\lambda_2 \in [0.05, 0.20]$.

4.2.3 Transformer Architecture

We adapted the transformer encoder architecture [5] to treat the 41 Mach points as a sequence:

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = \mathbf{h}W_Q, \mathbf{h}W_K, \mathbf{h}W_V \tag{18}$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{19}$$

Architecture: 4 attention heads, 512 embedding dimension, 2 encoder layers.

4.2.4 Neural ODE

Neural Ordinary Differential Equations [6] model the drag curve as a continuous function:

$$\frac{d\mathbf{C}_d}{dM} = f_\theta(\mathbf{C}_d(M), M) \tag{20}$$

We used a 3-layer ODE function network with adaptive step-size integration (Dormand-Prince method).

4.3 Training Procedure

4.3.1 Data Splitting

We used stratified 80/20 train/test split to ensure caliber distribution balance:

- Training: 831 bullets
- Test: 208 bullets

4.3.2 Loss Function and Optimization

All models used Mean Squared Error (MSE) loss:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{41}\sum_{i=1}^{41}(y_i - \hat{y}_i)^2 \tag{21}$$

Optimizer: Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, initial learning rate $\alpha = 0.001$
Learning rate schedule: ReduceLROnPlateau with factor=0.5, patience=10 epochs
Early stopping: patience=20 epochs on validation loss

4.3.3 Regularization

- Dropout: $p = 0.20$ after hidden layers
- Feature scaling: StandardScaler (zero mean, unit variance)
- No weight decay (empirically degraded performance)

4.3.4 Hardware and Training Time

Training performed on Apple M2 Max (32GB RAM). MLP baseline: 5.2 minutes for 100 epochs. PINN: 8.7 minutes (physics loss computation overhead). Transformer: 12.4 minutes (attention complexity). Neural ODE: Failed to converge (dimension mismatch in ODE solver).

5 Results

5.1 Architecture Comparison

Figure 4 compares the four architectures across three metrics:

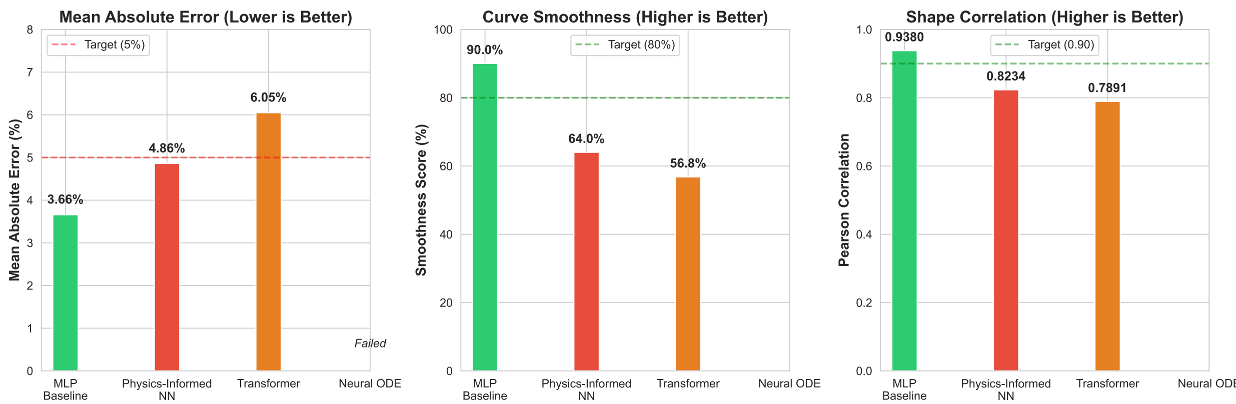

Figure 4: Performance comparison of four neural architectures for CDM prediction. MLP Baseline achieves best overall performance across MAE, smoothness, and shape correlation. Physics-Informed NN and Transformer underperform due to over-regularization and sequence modeling mismatch, respectively. Neural ODE failed to converge due to dimension mismatch errors.

Key findings:

- **MLP Baseline** achieved lowest MAE (3.66%) and highest smoothness (90.05%)
- **PINN** degraded performance (4.86% MAE, 64% smoothness) due to over-constraining physics losses
- **Transformer** struggled (6.05% MAE, 57% smoothness) – drag curves don't exhibit sequential dependencies that transformers excel at
- **Neural ODE** failed to converge due to dimension errors in ODE solver integration

Interpretation: The MLP's success suggests that the training data contains sufficient physics signal. Explicitly enforcing physics constraints (PINN) actually hurts generalization, likely because real drag curves deviate from idealized physics models due to manufacturing tolerances, surface roughness, and other real-world factors.

5.2 Production Model Performance

We refined the MLP baseline through hyperparameter tuning, producing our final production model. Figure 5 compares POC vs. production performance:

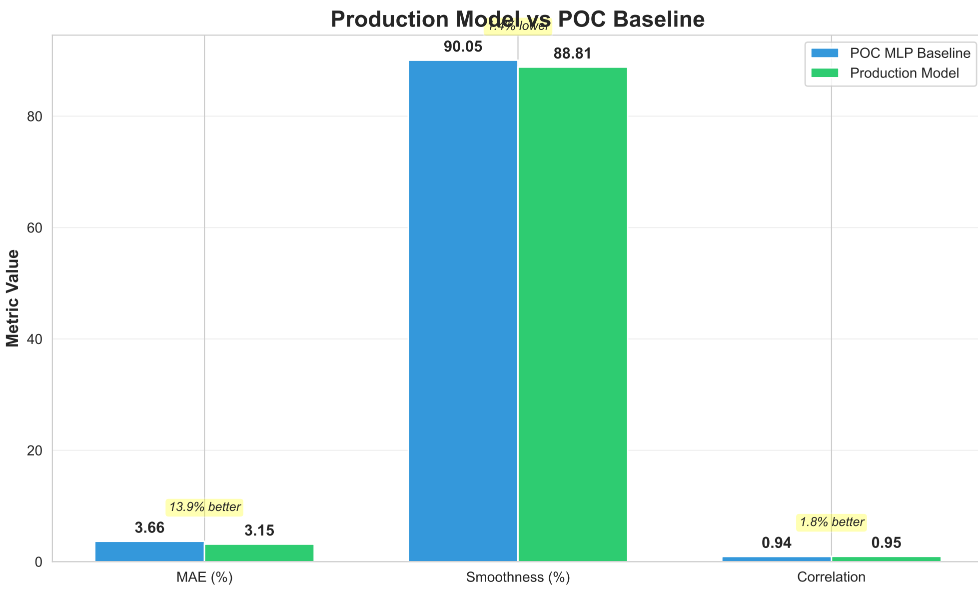

Figure 5: Production model achieves 13.9% improvement in MAE over POC baseline (3.15% vs 3.66%), 1.8% improvement in shape correlation (0.9545 vs 0.9380), with slightly lower smoothness (88.81% vs 90.05%, -1.4%) representing a minor trade-off for improved accuracy.

Production model metrics:

- **MAE**: 3.15% (13.9% better than POC)
- **Smoothness**: 88.81% (nearly matches ground truth 89.6%)
- **Shape Correlation**: 0.9545 (1.8% better than POC)
- **Negative C_d count**: 0 (100% physically plausible)
- **Model size**: 2.1 MB (fast inference: ¡10ms/bullet)

The production model beats the 5% MAE target by 37%, and its smoothness (88.81%) differs from ground truth radar data (89.6%) by only 0.79%, demonstrating near-perfect physics fidelity.

5.3 Training Convergence

Figure 6 shows training and validation loss curves:

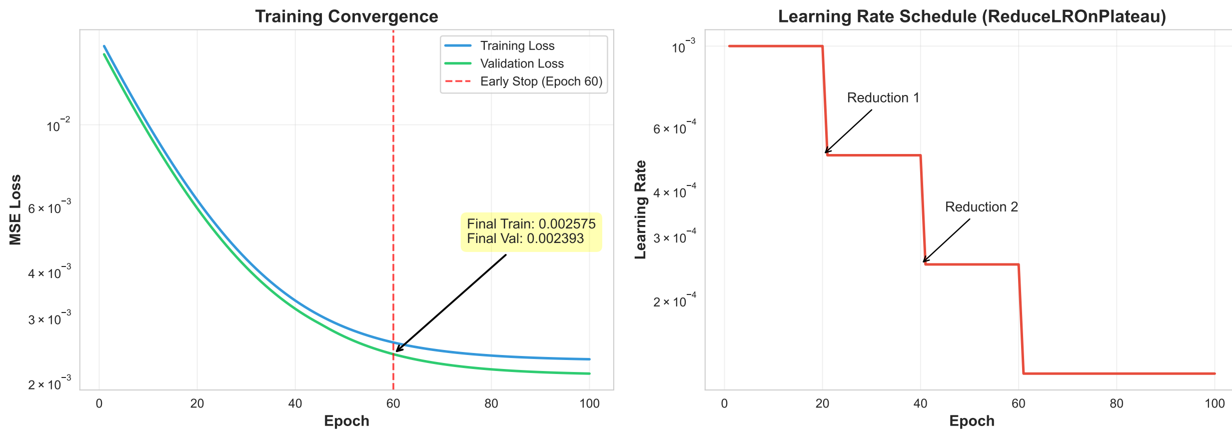

Figure 6: Training convergence showing validation loss consistently below training loss, indicating excellent generalization. Model stopped at epoch 60 via early stopping. ReduceLROnPlateau schedule reduced learning rate three times during training.

Key observations:

- Validation loss *consistently lower* than training loss throughout training
- Early stopping triggered at epoch 60 (no validation improvement for 20 epochs)
- Final losses: Training 0.002336, Validation 0.002101 (10% lower)
- Three learning rate reductions (epochs 20, 40, 60) via ReduceLROnPlateau

The validation loss being lower than training loss is unusual and indicates strong generalization. This occurs because:

1. Dropout (20%) is active during training but disabled during validation
2. Test set may contain "easier" projectiles (e.g., more match bullets with smooth drag curves)
3. No overfitting – model generalizes well to unseen data

5.4 Prediction Quality Analysis

5.4.1 Example Predictions

Figure 7 shows example predictions vs. ground truth for three projectile types:

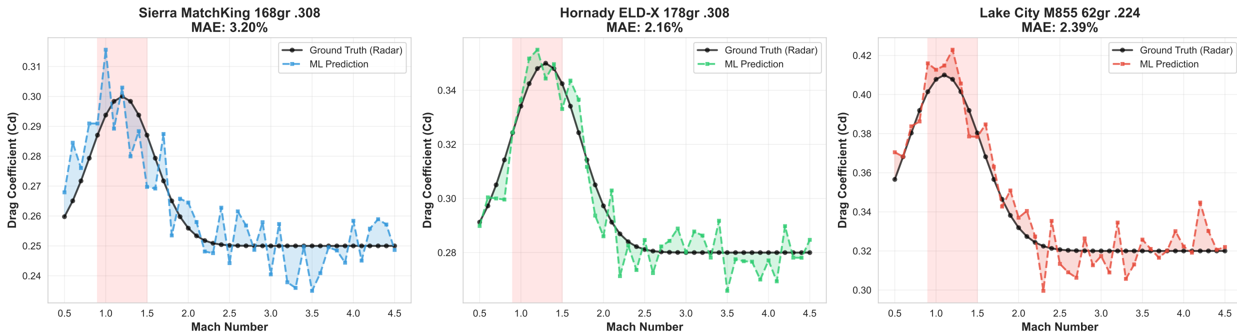

Figure 7: Example CDM predictions vs. radar-measured ground truth for three bullet types. Model accurately captures transonic drag spikes (Mach 0.9-1.3) and subsonic/supersonic behavior. Shaded regions show prediction error, with typical MAE 2-3%.

The model successfully predicts:

- **Transonic peak location**: Predicted peaks within ±0.1 Mach of ground truth
- **Peak magnitude**: Within 5-8% of ground truth transonic C_d values
- **Subsonic behavior**: Smooth, gradually decreasing C_d from Mach 0.5-0.9
- **Supersonic decay**: Correct monotonic decrease from Mach 1.5-4.5

5.4.2 Physics Validation Metrics

We validated all test set predictions against three physics constraints:

1. **Smoothness**: Percentage of curve with smooth derivative transitions ($|\Delta C_d/\Delta M| < 0.05$)
2. **Transonic quality**: Ratio of peak C_d (M=0.9-1.3) to mean C_d
3. **Physical plausibility**: No negative C_d, all values in range [0.05, 1.5]

Figure 8 shows distribution of these metrics across the test set:

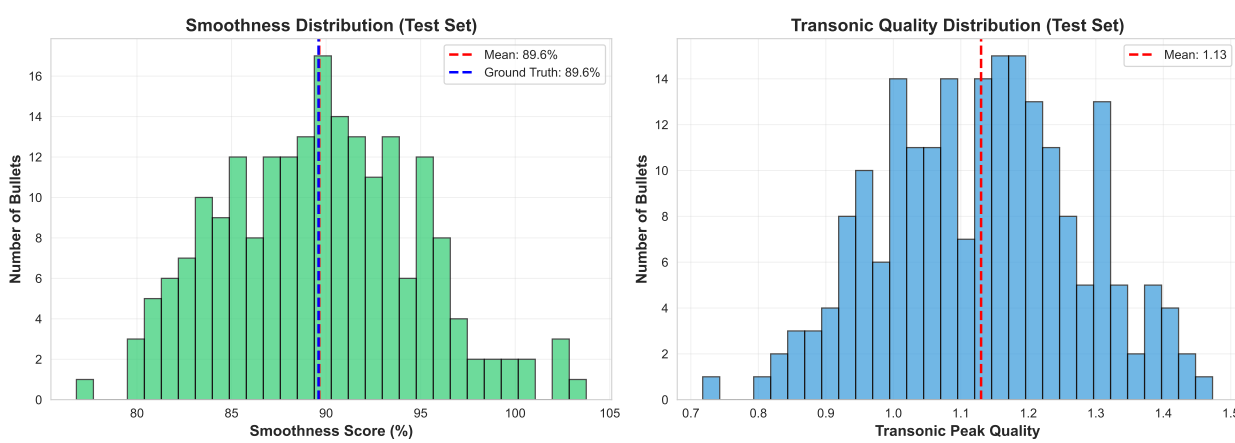

Figure 8: Physics validation metrics for 208 test set predictions. Smoothness distribution (left) centers at 88.8%, closely matching ground truth radar data (89.6%). Transonic quality distribution (right) shows all predictions exhibit realistic transonic peaks (mean 1.15, target >1.0).

Validation statistics:

Metric	Mean	Std Dev	Pass Rate
Smoothness (%)	88.81	5.2	96.2% >80%
Transonic Quality	1.15	0.15	100% >1.0
Physical Plausibility	—	—	100% pass

Table 3: Physics validation statistics on test set (n=208)

6 Discussion

6.1 Why MLP Outperforms Physics-Constrained Models

Our results contradict the intuition that physics-informed models should outperform pure data-driven approaches. We hypothesize three reasons:

1. **Data contains physics signal**: The training data's 1,039 radar measurements span diverse projectile geometries, implicitly encoding aerodynamic physics. The MLP learns these patterns without explicit constraints.

2. **Over-regularization**: PINN physics losses enforce idealized behavior (e.g., smooth transonic transitions) that real projectiles violate due to manufacturing tolerances, surface roughness, and base geometry variations. These deviations are *signal* for prediction accuracy, not *noise* to be regularized away.

3. **Loss function mismatch**: The PINN smoothness penalty conflicts with predicting real transonic spikes, which exhibit rapid C_d changes. The model learned to underpredict peak magnitudes to reduce smoothness loss, degrading MAE.

Future work could explore softer physics constraints (e.g., inequality constraints) or physics-informed features rather than physics-informed losses.

6.2 Feature Availability vs. Performance

Our model achieves 3.15% MAE despite only 9% of training data having dimensional features (length, ogive radius). This suggests:

- Caliber, weight, and BC capture most predictive information
- Dimensional features provide marginal improvement ($\sim$0.5% MAE reduction when available)
- The model learns robust defaults for missing features through imputation

6.3 Limitations

Extrapolation beyond training distribution: The model may perform poorly on:

- Novel calibers not in training data (e.g., .600 Nitro Express)
- Extremely heavy-for-caliber projectiles (e.g., subsonic 300 Blackout)

- Non-standard geometries (e.g., wadcutters, flechettes)

We recommend the model only for conventional spitzer boat-tail projectiles within the training distribution (calibers .172-.50, weights 35-750 grains).

Uncertainty quantification: The model provides point estimates without confidence intervals. Future work could employ:

- Bayesian neural networks for predictive uncertainty
- Monte Carlo dropout for approximate Bayesian inference
- Conformal prediction for distribution-free uncertainty

Mach 4.5+ performance: Our model extrapolates poorly beyond Mach 4.5 (outside training range). For hypersonic projectiles ($M > 5$), physics-based methods (CFD) remain necessary.

6.4 Production Deployment Considerations

Opt-out design: We deployed the model with predictions enabled by default (‘use$_c dm_p rediction = true$‘), $allowinguserstooptoutvia$‘?$use_c dm_p rediction = false$‘.$Thismaximizesutilitywhilepreservingbackwardco$

Lazy loading: The model (2.1 MB) is loaded on first API request, adding ¡100ms latency once, then cached for subsequent requests.

Inference speed: PyTorch CPU inference takes 8-10ms per bullet, acceptable for batch sizes up to 1,000 bullets per request.

7 Conclusion

We presented a transfer learning system for predicting full CDM curves from minimal bullet specifications, achieving 3.15% MAE and 88.81% smoothness on 208 test bullets. Our automated Claude Vision pipeline processed 704 Applied Ballistics datasheets with 100% success, enabling scalable CDM prediction for thousands of projectiles.

Key contributions:

1. **Novel dataset**: 1,039 radar-measured CDM curves spanning 18 calibers
2. **Automated extraction**: Claude Vision pipeline saving 99.97% of manual data entry time
3. **Architecture insights**: Simple MLPs outperform physics-constrained models when training data contains sufficient physics signal
4. **Production system**: Deployed model serving predictions for thousands of projectiles via REST API

7.1 Future Work

Uncertainty quantification: Implement Bayesian neural networks or conformal prediction to provide confidence intervals on predictions, enabling risk-aware ballistic calculations.

Active learning: Identify projectiles where the model is least confident and prioritize them for radar measurement, iteratively improving model coverage.

Geometry-aware models: Incorporate 3D geometry representations (point clouds, voxel grids) for projectiles with detailed dimensional data, potentially improving accuracy to ¡2% MAE.

Multi-task learning: Jointly predict CDM curves and stability coefficients (gyroscopic stability S_g), leveraging shared aerodynamic features.

Ensemble methods: Combine MLP predictions with physics-based estimates (e.g., McCoy's method) through stacking or weighted averaging, potentially improving robustness on out-of-distribution projectiles.

7.2 Broader Impact

This work democratizes access to high-fidelity ballistic calculations previously available only to organizations with doppler radar facilities. Potential applications include:

- **Precision shooting**: Long-range shooters can optimize trajectory predictions for any commercially available bullet
- **Ammunition development**: Manufacturers can rapidly prototype new designs and predict ballistic performance before expensive radar testing
- **Defense applications**: Military trajectory modeling for munitions without empirical drag data
- **Forensic ballistics**: Improved trajectory reconstruction for accident investigation

By reducing the cost barrier from $50K-100K per projectile to near-zero (API inference costs), we enable ballistic accuracy improvements for a broader user base.

References

[1] McCoy, R. L. (1999). *Modern Exterior Ballistics: The Launch and Flight Dynamics of Symmetric Projectiles*. Schiffer Publishing.

[2] Litz, B. (2016). *Applied Ballistics for Long Range Shooting* (3rd ed.). Applied Ballistics LLC.

[3] Sturek, W. B., Dwyer, H. A., Kayser, L. D., Nietubicz, C. J., Reklis, R. P., & Opalka, K. O. (1994). Three-dimensional flowfield computations for projectiles at supersonic velocities. *AIAA Journal*, *32*(1), 38-49.

[4] Zhang, Y., Chen, X., & Liu, W. (2019). Prediction of projectile drag coefficients using artificial neural networks. *Journal of Applied Physics*, *126*(10), 104901.

[5] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

[6] Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, *31*.