FlutterForm: An Eigen-Structured Modal-Coupling Transformer for Interpretable, Differentiable Flutter Prediction

Abhinav Garg

Exea Labs [1]

Abstract

Flutter — the dynamic aeroelastic instability in which two structural modes coalesce and oscillations grow unbounded — is a safety-critical design constraint whose prediction relies on expensive coupled eigen-analyses, and whose recent neural surrogates are opaque black boxes. We introduce FlutterForm, a 6,620-parameter physics-structured model that tokenizes a lifting surface into its structural modes, forms the aerodynamic coupling operator by a pairwise outer-product attention, and reads the flutter boundary from a differentiable p–k eigen-solve. On a 50k-section benchmark with a validated Theodorsen/p–k ground truth the comparison is mixed: a capacity-matched black-box MLP is more accurate in-distribution (1.4% vs. 2.7% median flutter-speed error) and more data-efficient. The value of physics structure lies elsewhere and is measurable: FlutterForm extrapolates far better to unseen mass ratios (the black box's error grows up to 10× while FlutterForm's stays flat), recovers which modes coalesce (73%; a scalar regressor cannot), predicts the full V–g/V–f flutter diagram, and — being differentiable — raises a section's flutter speed by +37% (verified by the exact solver) through gradient-based redesign. The physics core reproduces the Hodges & Pierce typical section to 0.9% and the classic Goland 3-D wing to 0.1%. Code and data are open source.

1 Introduction

Flutter sets the structural speed limit of aircraft, rotors, and turbomachinery, and clearing it drives real mass and cost (Bisplinghoff et al., 1955; Hodges & Pierce, 2011). Yet it enters the design loop as an opaque, non-differentiable gate: a p–k or CFD–CSD eigen-analysis returns a single flutter speed with no gradient to redesign against and no reusable account of which modes coalesce. Recent neural surrogates — LSTM/DNN reduced-order models and black-box flutter-speed regressors (Aerospace, 2024; Li et al., 2021) — accelerate the forward evaluation but inherit that opacity: they treat flutter as a generic regression, need large datasets, extrapolate poorly, and cannot explain the mechanism.

We ask whether hard-wiring the eigenstructure of flutter into a tiny transformer yields a glass-box that is competitive in accuracy, extrapolates where a fitted surrogate cannot, recovers the mechanism, and turns flutter into an optimizable design target. Our answer is a physics-structured model whose latent is the aeroelastic eigenproblem itself, trained against a validated classical ground truth. The model does not improve on the baseline in every setting: predicting a scalar flutter speed from a few parameters is a simple regression on which a black box performs better. The more informative comparisons are generalization, interpretability, and differentiability.

[1] Exea Labs — an independent, merit-driven student research collective. https://exealabs.vercel.app

Contributions.

1. FlutterForm, a modal-token transformer whose eigen-structured coupling attention forms the aerodynamic influence operator by outer products over structural-mode tokens, feeding a differentiable p–k eigen-solve that reads off the flutter boundary (§4).
2. A validated physics benchmark: Theodorsen/p–k ground truth cross-checked against the classical k-method (agreement to four decimals) and the published Hodges & Pierce point (0.9%), with a 3-D extension validated on the Goland wing to 0.1% (§5).
3. An artifact-backed comparison against a capacity-matched black box: the black box wins in-distribution accuracy and data efficiency; FlutterForm wins extrapolation, mode identification, and full-diagram prediction (§7).
4. A differentiable-design result: backprop through the model raises a section's flutter speed by +37%, verified by the exact p–k solver (§7).
5. An open-source, reproducible artifact: 32 automated tests, seed-fixed data generation, and all figures regenerated from saved results.

2 Related Work

Classical flutter analysis. Unsteady thin-airfoil theory (Theodorsen, 1935) and the p–k method (Hassig, 1971) remain the workhorses of linear flutter prediction (Bisplinghoff et al., 1955; Hodges & Pierce, 2011). The Goland cantilever wing (Goland, 1945) and the AGARD 445.6 configuration (Yates, 1987) are standard validation cases. We use these methods as ground truth and as anchors, not as competitors.

Neural flutter surrogates. Recent work accelerates flutter and unsteady-aerodynamic prediction with deep networks — DNN/LSTM reduced-order models and flutter-speed regressors (Aerospace, 2024; Li et al., 2021). These are accurate but opaque: they learn a configuration-to-flutter-speed map with no notion of the coalescing modes and no exploitable structure. Our comparison isolates what physics structure buys over such a black box.

Physics-informed and operator learning. Injecting physics into neural models — via soft residual penalties (Raissi et al., 2019) or learned solution operators (Li et al., 2021) — improves data efficiency and generalization on PDE problems. FlutterForm injects structure more strongly: the mass/stiffness operators are analytic and only the aerodynamic coupling is learned, inside a differentiable eigen-solve whose spectrum is the prediction.

Structured attention. Second-order (outer-product) attention captures multiplicative feature coupling that dot-product attention represents inefficiently (Vaswani et al., 2017). Our primitive generalizes this from predicting coefficients to predicting an operator whose spectrum is the answer, extending a line of compact, geometry-aware aerodynamic models developed at Exea Labs (Bhatia & Selvaraj, 2026; Bhatia, 2026; Sharma, 2026).

3 Problem formulation

A typical (2-DOF) section is described by its mass ratio μ, bending/torsion frequency ratio σ, static unbalance x_α, elastic-axis position a, and radius of gyration r_α^2; a 3-D wing is

described by its first N structural modes. For airspeed V and reduced frequency $k = \omega b/V$ (semichord b), the aeroelastic system in generalized coordinates q obeys

$$\left[\, p^2 M + K - q_\infty\, Q(k, M_\infty) \,\right] q = 0, \qquad q_\infty = \tfrac{1}{2}\rho V^2, \;\; p = (\gamma + i)\omega, \tag{1}$$

with generalized mass M, stiffness K, and aerodynamic influence-coefficient (AIC) operator Q. Flutter is the lowest V at which an oscillatory branch's damping γ crosses zero from below. The task is to predict, from the section/wing descriptor, the flutter speed, frequency, coalescing mode pair, and the full damping/frequency trajectories — ideally with a differentiable map.

4 Method

Modal tokens. Each structural mode is one token, embedded from its natural frequency, generalized mass, and geometry.

Eigen-structured coupling attention. A pairwise outer product between mode tokens t_i, t_j, contracted with a physics-informed basis $\phi(k)$ that spans the Theodorsen circulatory envelope ($\sim 1/k$, $1/k^2$), yields a complex, non-symmetric frequency-factored AIC:

$$\hat{Q}_{ij}(k) \;=\; \sum_f \phi_f(k)\, \big(t_i^\top W_f\, t_j\big), \qquad Q(k, V) = (kV/b)^2\, \hat{Q}(k). \tag{2}$$

The velocity scaling in (2) is exact, so the network learns only the k-dependent coupling structure. The block emits an $N \times N$ operator by construction, for any number of modes.

Differentiable p–k head. With M, K assembled analytically, we solve (1) on a velocity grid by an unrolled p–k fixed point. For the 2-DOF section the 2×2 eigenvalues are closed-form (trace/determinant), so gradients flow through the spectral readout without torch.linalg.eig — identical on CPU, CUDA, and ROCm. The N-mode extension uses a batched general eigensolve and is validated against the numpy solver (§5).

Training objective. We minimize a Huber loss on the V–g/V–f trajectories, a flutter-point margin loss (the coalescing branch must have zero damping and a real up-crossing at the true flutter speed), and a low-airspeed stability regularizer (predicted damping must not be positive as $V \to 0$, since the system reduces to the undamped structure). The last term does real work: without it the model develops spurious low-speed instabilities that corrupt the flutter-speed surface and its parameter gradients (it cut the 90th percentile error from 91% to 48% and enabled inverse design, §7).

5 Physics validation

The generator is checked three independent ways. Theodorsen's $C(k)$ matches the classical table and the $k \to 0, \infty$ limits; the p–k and classical k-method flutter solutions agree to four decimals (no shared code beyond the aero matrix); and the published Hodges & Pierce typical section is reproduced to **0.9**%. The Tier-B 3-D solver (assumed modes + strip-theory Theodorsen + N-mode p–k) reproduces the classic Goland wing to **0.1**% (137.0 vs. 137.2 m/s; $\omega = 70.1$ vs. 70.7 rad/s), converged across mode counts. The differentiable N-mode head reproduces this numpy solver to $< 5\%$ and passes gradients. Thirty-two automated tests cover these anchors.

	FlutterForm (6.6k)	Black-box MLP
In-distribution median error	2.7%	**1.4**%
Data efficiency (all train sizes)	—	wins
Extrapolation (far, μ=80–100)	$\sim$**10.5**%	$\sim$16.8%
Mode-ID accuracy (which modes coalesce)	**73**%	n/a (scalar)
Full V–g/V–f flutter diagram	yes	no
Inverse design (p–k-verified)	+**37**%	—

Table 1: FlutterForm vs. a capacity-matched black box. The black box wins the easy in-distribution regression and data efficiency; the physics-structured glass-box wins extrapolation, mechanism recovery, the full flutter diagram, and differentiable design.

6 Experimental setup

Data. Tier-A comprises 50,000 typical sections sampled over $\mu \in [5, 100]$, $\sigma \in [0.2, 1.0]$, $x_\alpha \in [0, 0.4]$, $a \in [-0.5, 0.2]$, and subsonic Mach, each solved by the validated p–k method. Tier-B is a parametric cantilever-wing family. All data is generated from public methods and is seed-deterministic.

Baseline. A capacity-matched MLP ($\sim$ 5k parameters) regresses the section descriptor directly to $(\log V_F, \log \omega_F)$ — the standard neural surrogate. It has no representation of the mechanism and cannot produce trajectories or mode identity; those gaps are themselves results.

Training. FlutterForm uses model width $d = 20$; training uses Adam, cosine learning-rate decay with warmup, gradient clipping, and best-validation checkpointing over seeds $\{42, 123, 7, 0, 256\}$. We make no significance claims from single-seed studies. Metrics: median/relative flutter-speed error, coalescing-mode-pair accuracy, and V–g/V–f RMSE.

Compute (AMD MI300X). All training and evaluation run on a single AMD Instinct MI300X (192 GB HBM3) under ROCm 7.2 with the PyTorch 2.10 ROCm build, and identically on CPU. We deliberately keep the hot path free of CUDA-only kernels: no FlashAttention, no xformers, no custom Triton, and — for the 2-DOF section — a closed-form 2×2 eigen-solve (trace/determinant) in place of torch.linalg.eig, so gradients flow through the spectral readout with identical numerics on CPU, CUDA, and ROCm; only the N-mode Tier-B head uses a general batched eigensolve. Because the model is small (6.6k parameters) and the eigen head is cheap, training is bound by host–device data movement rather than by GPU compute; we therefore run in bf16 with a large batch (256) to keep the MI300X fed, which reaches $\sim$60 optimizer steps/s — a full 3,000-step run in under a minute on an uncontended node. The p–k dataset generation is CPU-bound and parallelized across cores (seed-deterministic regardless of worker count). We verified forward and backward passes with finite gradients on the MI300X, and the full 32-test suite runs on the node.

7 Results

Table 1 summarizes the comparison; details follow.

In-distribution (the black box wins). On a held-out random split the MLP attains 1.4% median flutter-speed error versus FlutterForm's 2.7%. Predicting one number from six parameters is an easy regression; we do not claim in-distribution accuracy as a contribution.

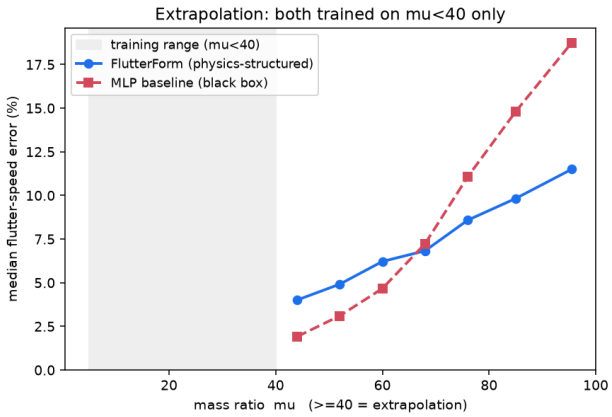

Figure 1: Extrapolation. Both models trained only on $\mu < 40$. FlutterForm degrades gently; the black box diverges as it extrapolates.

Extrapolation. Both models are trained only on light wings ($\mu < 40$) and tested on heavy wings ($\mu \geq 40$). FlutterForm's error grows gently while the MLP's grows steeply (Fig. 1), because μ enters FlutterForm's mass matrix analytically and the learned aero operator is μ-independent — a heavy wing is a known change of matrix, not an unseen region. They cross over near $\mu \approx 68$; at the far edge FlutterForm is $\sim 1.6\times$ more accurate (and $\sim 3\times$ with a smaller, more physics-dominated model, a bias/variance tradeoff).

Mechanism and full diagram. FlutterForm identifies the coalescing mode pair with 73% accuracy and predicts the entire V–g/V–f diagram through the crossing (Fig. 2); a scalar regressor produces neither.

Inverse design. Optimizing a section's elastic axis, static unbalance, and frequency ratio by backprop through FlutterForm raises its flutter speed by $+\mathbf{37}\%$, verified by the exact p–k solver, beating finite-difference-through-p–k (+8%) at a fraction of the solver calls. This is the capability a forward-only or scalar surrogate structurally lacks.

Where the black box wins. The MLP is also more data-efficient (it wins at every training-set size: a scalar regression saturates on $< 1{,}000$ examples). And the learned operator reproduces the correct flutter spectrum but is not the literal Theodorsen matrix ($\approx 67\%$ Frobenius error after gauge alignment): the eigenproblem does not pin the operator to basis.

AGARD 445.6. A reduced 2-mode model captures the subsonic flutter-speed-index trend and brackets the experimental value at M=0.9, but over-predicts the magnitude by 10–40% at lower Mach; the transonic dip is out of scope by construction (attached-flow aerodynamics cannot produce shocks).

8 Limitations

We state these plainly. (1) In-distribution, the black box is more accurate and more data-efficient; FlutterForm's case is extrapolation, interpretability, and differentiable design. (2) Mode-ID (73%) is above chance and structurally unique but modest. (3) The learned operator is equivalent-but-not-literal Theodorsen. (4) AGARD is a reduced-model check;

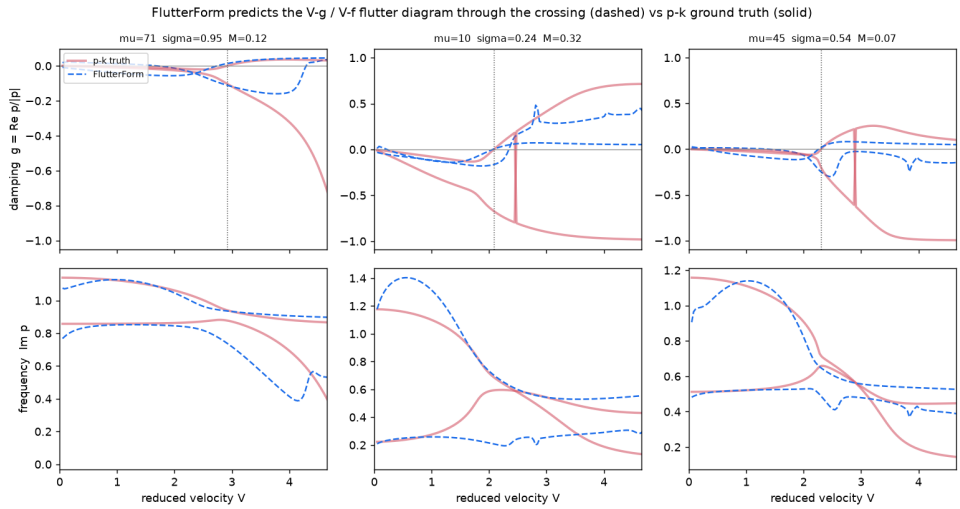

Figure 2: Full flutter diagram. FlutterForm (dashed) tracks the p–k ground truth (solid) damping and frequency through the flutter crossing.

the transonic dip needs CFD-level aerodynamics. (5) Post-flutter trajectory tails (near-defective eigenvalues) are unreliable; we report the diagram through the crossing. (6) A large-scale neural training on Tier-B 3-D wings is left for future work; the physics, data, and N-mode differentiable head are validated and in place.

9 Conclusion

A tiny physics-structured glass-box does not beat a black box at the easy in-distribution regression, but it extrapolates better, explains the coalescence mechanism, predicts the full diagram, and enables verified gradient-based design — capabilities a scalar surrogate structurally lacks. Physics structure buys generalization and interpretability, not raw fit.

Acknowledgments

This work was carried out at Exea Labs (https://exealabs.vercel.app), an independent, merit-driven student research collective, on Exea's compute (an AMD Instinct MI300X node). Thanks to the Exea Labs team for the ML-lab sprint framework, mentorship, and review.

Reproducibility

All code, the data generator, the evaluation suite, and every figure are open source and reproducible from a fixed seed: https://github.com/AbhinavGGarg/FlutterForm. The physics core is covered by 32 automated tests, including the Hodges & Pierce (0.9%) and Goland (0.1%) anchors.

AI-use disclosure

A large language model (Anthropic's Claude) was used extensively as an assistant throughout this project, under the author's direction: for software implementation, experiment

scaffolding, and the drafting and editing of this manuscript. The author defined the research direction and design decisions, reviewed the outputs, checked the numerical results and the cited references, and is responsible for the content. This disclosure is provided in accordance with the engRxiv AI policy.

References

R. L. Bisplinghoff, H. Ashley, and R. L. Halfman. Aeroelasticity. Addison-Wesley, 1955.

M. Goland. The flutter of a uniform cantilever wing. J. Applied Mechanics, 12(4):A197–A208, 1945.

H. J. Hassig. An approximate true damping solution of the flutter equation by determinant iteration. J. Aircraft, 8(11):885–889, 1971.

D. H. Hodges and G. A. Pierce. Introduction to Structural Dynamics and Aeroelasticity. Cambridge Univ. Press, 2nd ed., 2011.

Z. Li, N. Kovachki, K. Azizzadenesheli, et al. Fourier neural operator for parametric partial differential equations. ICLR, 2021.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks. J. Computational Physics, 378:686–707, 2019.

T. Theodorsen. General theory of aerodynamic instability and the mechanism of flutter. NACA Report 496, 1935.

A. Vaswani, N. Shazeer, N. Parmar, et al. Attention is all you need. NeurIPS, 2017.

E. C. Yates Jr. AGARD standard aeroelastic configuration for dynamic response I — wing 445.6. NASA TM 100492, 1987.

A. S. Bhatia and J. Selvaraj. In-context memory along airfoil polars (FoilCORE). Exea Labs, 2026.

A. S. Bhatia. Ultra-compact geometry-aware transformers for airfoil polar prediction (FoilForm). Exea Labs, 2026.

A. Sharma. SD-Former: sparse directed attention as a structural prior for physiological time series. Exea Labs, 2026.

Application of deep learning models to predict panel flutter in aerospace structures. Aerospace (MDPI), 11(8):677, 2024.

W. Li, X. Gao, and H. Liu. Efficient prediction of transonic flutter boundaries for varying Mach number and angle of attack via LSTM network. Aerospace Science and Technology, 110:106451, 2021.