WHITE PAPER · METHODS

Runtime-Safe Synthesis of Engineering Stress–Strain Curves

From material metadata and predicted helper properties, through a constrained empirical reconstruction stage.

WORKING TITLE **Curve Synthesizer** · SURFACES MaterialPoint, MCP

ABSTRACT

The Curve Synthesizer is a hybrid materials-informatics tool for generating plausible engineering stress–strain curves when a measured curve is unavailable at runtime. The system is designed around a strict separation between runtime-safe inputs and offline-only labels. It first predicts a compact set of mechanically meaningful helper properties from structured material metadata and context, then reconstructs a full curve through an empirical synthesis stage, and finally can embed the result into lightweight constitutive and finite-element-compatible downstream stages for validation and simulation readiness. The purpose of the tool is not to replace laboratory testing or to claim discovery of a constitutive law. Its role is to provide a controlled, auditable approximation that is useful for early-stage engineering workflows, simulation preparation, and data completion where direct curve measurements are missing or sparse.

AT A GLANCE

runtime-safe inputs, not measured curves · helper layer, not opaque end-to-end · engineering approximation, not constitutive discovery

STATUS

Draft for internal discussion. This document is intentionally descriptive rather than fully procedural. It explains the idea, scope, and validation posture of the Curve Synthesizer, but omits implementation

> details that would enable straightforward copying or reconstruction of the production pipeline.

1 Problem Statement

Engineering workflows often need a full stress–strain curve, but available information is usually fragmented. In many practical settings, teams may have:

- a material identity and formulation metadata;
- a conditioning state and temperature;
- a limited set of scalar properties from datasheets or structured databases;
- no directly measured curve at the requested runtime condition.

This creates a gap between what simulation tools require and what enterprise materials records typically provide. The Curve Synthesizer addresses that gap by converting runtime-safe structured inputs into a predicted curve representation, while maintaining a clear boundary between inference inputs and offline validation references.

2 Design Goal

The design goal is not "curve generation from anything available." The design goal is narrower:

- use only runtime-safe structured inputs at inference time;
- predict a physically interpretable helper layer before predicting the full curve;
- preserve traceability across condition selection, feature construction, helper routing, curve synthesis, and optional FEM embedding;
- keep measured curves and offline derivations out of runtime inference;
- make the output useful for engineering review rather than presenting it as ground truth.

3 Conceptual Architecture

At a high level, the tool can be described as a staged stack:

1. runtime-safe feature assembly;
2. helper-property inference;
3. regime classification;
4. empirical curve synthesis;
5. optional constitutive fitting and FEM-oriented embedding;
6. offline comparison against measured references when such references exist.

This separation matters. It reduces the risk of hidden label leakage, makes the intermediate predictions interpretable, and allows engineering teams to inspect whether failure arises from metadata sparsity, helper prediction, synthesis logic, or downstream constitutive transfer.

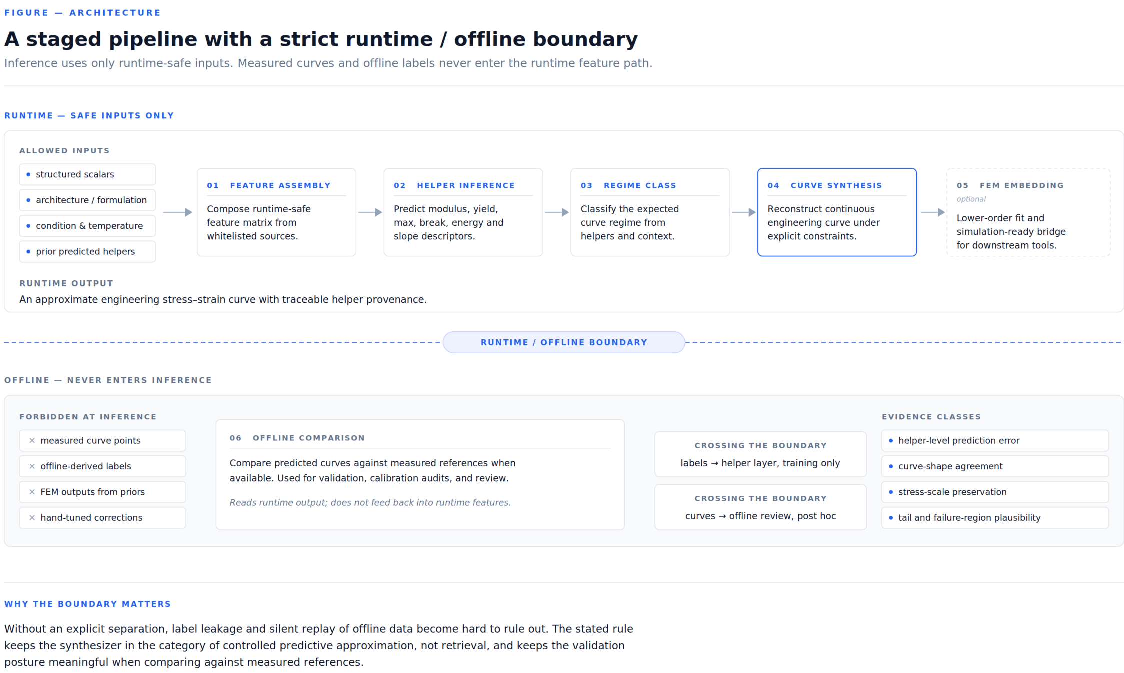

FIGURE 1

A staged pipeline with a strict runtime / offline boundary. Inference uses only structured, runtime-safe inputs; measured curves and offline labels are kept outside the runtime feature path and used only for validation.

4 Input and Output Boundary

The most important governance rule in the current design is the runtime boundary.

Allowed runtime inputs include structured scalar properties from whitelisted sources, architecture or formulation descriptors judged runtime-safe, condition labels and temperature transforms, and previously predicted intermediate quantities.

Forbidden runtime inputs include measured curve points, offline Curve Synthesizer labels and helper derivations, FEM outputs from prior runs, and material-specific hand-tuned corrections introduced outside the declared synthesis logic.

This boundary is central to the credibility of the tool. It keeps the synthesizer in the category of controlled predictive approximation rather than silent retrieval or replay of offline curve information.

5 Modeling Strategy

The modeling strategy is hybrid rather than end-to-end. This positioning is deliberate: prior reviews of neural-network constitutive modeling have shown that universal-function-approximator approaches can be powerful but are often hard to constrain, interpret, and audit [2], while physics-informed and structured approaches improve transparency and stability at the cost of additional design effort [3, 4].

5.1 Helper Layer

Instead of predicting the full curve directly from metadata, the system first predicts a set of helper quantities such as modulus-, yield-, maximum-, break-, energy-, and slope-related descriptors. This intermediate layer is attractive for three reasons. It is easier to inspect than a direct curve output. It supports structured consistency checks. And it gives engineering users a way to understand why a curve has a particular shape. Ensemble-style predictors with well-understood bias–variance properties are particularly well suited to this stage [1].

5.2 Regime-Aware Synthesis

The second stage uses the helper set and a predicted regime label to reconstruct a continuous engineering curve. In the present implementation family, this synthesis stage is empirical and constraint-aware, not a first-principles constitutive law. Smoothness, monotonicity, anchor consistency, and stress-scale preservation are treated as design constraints.

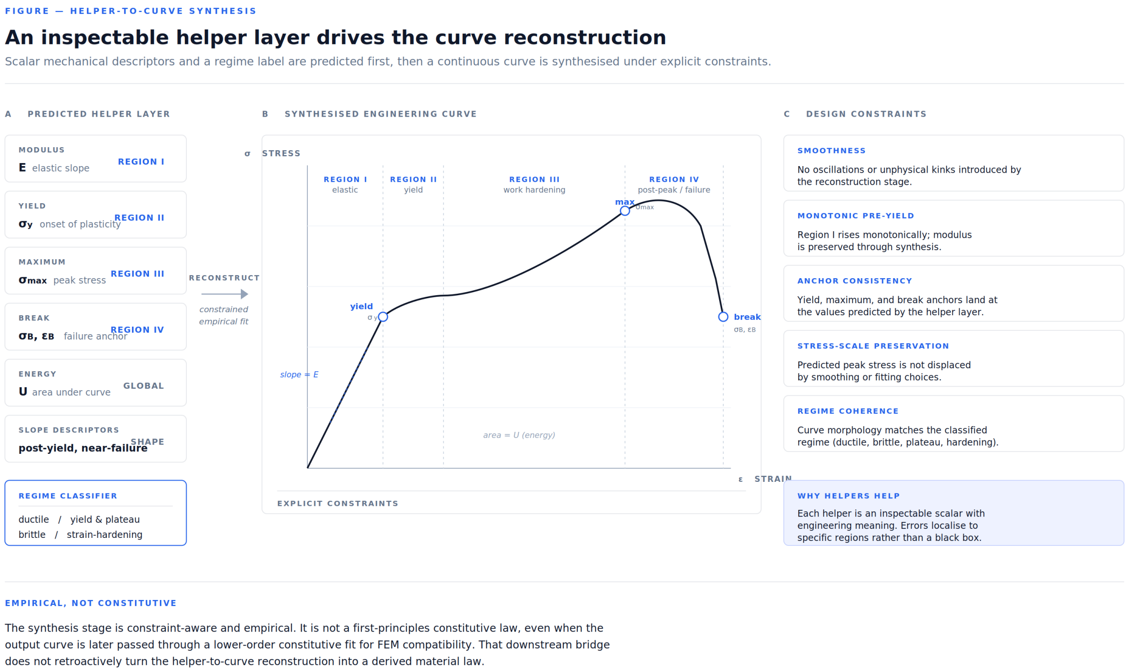

FIGURE 2

An inspectable helper layer drives the curve reconstruction. Scalar mechanical descriptors and a regime label are predicted first; then a continuous curve is synthesised under explicit smoothness, monotonicity, anchor-consistency, and stress-scale constraints.

5.3 Optional Constitutive Embedding

After synthesis, the generated curve can be passed through a lower-order constitutive fitting stage and then through a lightweight FEM-style runtime solver. This does not retroactively make the synthesis stage a constitutive model. Rather, it provides a compatibility bridge for downstream simulation workflows.

6 What the Tool Is and Is Not

The Curve Synthesizer should be described carefully.

It is a hybrid empirical reconstruction workflow; a helper-to-curve synthesis system; a runtime-safe approximation layer for missing curve data; and a bridge between structured materials records and downstream simulation assets.

It is not a replacement for mechanical testing; nor a universal constitutive law for polymers or metals; nor a proof that the predicted material response is physically unique; nor a substitute for validation under new loading modes, strain rates, or failure mechanisms.

7 Why This Structure Is Useful

The staged design provides practical advantages over a single opaque predictor.

First, it improves auditability. A user can inspect condition resolution, feature provenance, helper routing, regime probabilities, and curve-level artifacts separately.

Second, it is safer for deployment. The explicit schema boundary helps keep offline labels out of the runtime feature matrix.

Third, it is easier to validate. The team can compare helper errors, curve-shape errors, and downstream constitutive-fit behavior independently.

Fourth, it creates a better collaboration point with engineering experts. Subject-matter reviewers can debate helper plausibility and curve morphology without needing to inspect internal model weights.

8 Validation Philosophy

Validation should be framed as layered evidence rather than a single score. Relevant evidence classes include helper prediction error against offline labels, curve-shape agreement against measured references, stress-scale preservation, tail behavior and failure-region plausibility, smoothness and artifact screening, stability of optional constitutive fitting, and consistency of runtime behavior across temperatures and conditioning states.

This is a stronger posture than simply reporting one regression metric on resampled points. It also aligns better with engineering use, where a curve that is globally close but wrong at yield or failure may still be unusable.

9 Current Technical Posture

Based on the present codebase, the system already shows several strong design choices: explicit namespace separation between runtime inputs, predictions, and offline labels; documented semantic contracts for what is allowed at synthesis time; asset manifests and routing contracts for model bundles; group-aware train/test splitting by material identifier; and exported diagnostics that support review and traceability.

At the same time, the current posture should still be described as controlled internal tooling rather than settled scientific infrastructure. The implementation mixes experimental and production-adjacent paths, and the validation story is stronger in intent than in complete statistical coverage.

10 Known Limits and Caution Areas

The following limitations should be stated clearly in any public or partner-facing version of this paper:

- the synthesizer is only as reliable as the structured inputs and condition normalization;
- generalization outside the represented material families or temperature ranges may be poor;
- the curve synthesis stage is empirical even when later coupled to a constitutive fit;

- single-split or narrow holdout strategies can overstate confidence if family coverage is limited;
- prediction confidence and uncertainty quantification remain underdeveloped relative to the importance of the downstream decisions [5];
- outputs should be treated as engineering estimates subject to review, not authoritative material allowables.

11 Recommended Positioning

The most defensible positioning is:

> **POSITIONING STATEMENT**
>
> *The Curve Synthesizer is a runtime-safe, auditable approximation framework that predicts engineering stress–strain curves from structured material data through an intermediate helper-property layer. It is intended for early-stage engineering use, simulation preparation, and data completion, with offline validation against measured references where available.*

This framing is strong enough to communicate the value of the tool without overstating its scientific maturity. It is also consistent with the broader trajectory of materials informatics, which is moving toward auditable, evidence-anchored tooling rather than monolithic predictors [6].

12 Suggested Next Research Steps

The next steps that would most improve scientific credibility are to add uncertainty estimates at both helper and curve levels; expand holdout design to include stronger family-level and supplier-level stress tests; compare multiple synthesis branches under the same helper inputs and validation protocol; add calibration and error concentration analyses for yield, maximum stress, and break region behavior; and evaluate the downstream sensitivity of simulation outputs to curve synthesis error, not only the curve error itself.

13 Conclusion

The Curve Synthesizer represents a pragmatic middle ground between simple lookup systems and full constitutive discovery. Its main contribution is architectural: it formalises a runtime-safe path from structured material metadata to usable approximate curves, while preserving interpretability through a helper layer and keeping offline labels outside inference. That makes it valuable for engineering enablement, even if it should not yet be presented as a complete scientific replacement for measurement-based characterization.

REFERENCES

01 L. Breiman. Random Forests. *Machine Learning* **45**, 5–32 (2001).
link.springer.com/article/10.1023/A:1010933404324

02 J. Dornheim, L. Morand, H. J. Nallani, and D. Helm. Neural Networks for Constitutive Modeling – From Universal Function Approximators to Advanced Models and the Integration of Physics. arXiv:2302.14397 (2023).
arxiv.org/abs/2302.14397

03 E. Haghighat, S. Abouali, and R. Vaziri. Constitutive model characterization and discovery using physics-informed deep learning. arXiv:2203.09789 (2022).
arxiv.org/abs/2203.09789

04 K. Upadhyay, J. N. Fuhg, N. Bouklas, and K. T. Ramesh. Physics-informed Data-driven Discovery of Constitutive Models with Application to Strain-Rate-sensitive Soft Materials. arXiv:2304.13897 (2023).
arxiv.org/abs/2304.13897

05 J. Dai, S. Adhikari, and M. Wen. Uncertainty Quantification and Propagation in Atomistic Machine Learning. arXiv:2405.02461 (2024).
arxiv.org/abs/2405.02461

06 T. Lookman, Y. Liu, and Z. Gao. Materials Informatics: Emergence To Autonomous Discovery In The Age Of AI. arXiv:2601.00742 (2026).
arxiv.org/abs/2601.00742