MoDeVa 2.0 — What's New

MoDeVa is a single toolkit for the interpretable model lifecycle: data management, exploratory analysis, feature engineering, model training, interpretation and validation across accuracy, reliability, robustness and fairness. Version 2.0 keeps that surface unchanged and extends it in three directions. It adds four model families that read a fitted tree ensemble as a geometric or kernel object, it adds validation diagnostics that turn a single aggregate error number into a map of where and why a model fails, and it changes how the package is distributed and licensed. Every 1.x workflow — the `DataSet`, `ModelZoo`, `TestSuite`, `Experiment` and `Pipeline` APIs, and the existing model wrappers — runs without modification. The new material is additive.

The monograph documents each feature in full; chapter pointers appear at the end of this note.

1 New models

The four new model families share one premise. A fitted gradient-boosted ensemble is not only a predictor but a learned geometry: the leaves it routes a sample to encode how the model partitions feature space. DirectRS reshapes that geometry into a linear post-processor, the GBDT leaf kernel re-reads it as a kernel machine, ICL-MoE uses it as an embedding for local experts, and FuseKernel fuses it with other kernels. Each is a first-class MoDeVa estimator: it trains through the `ModelZoo`, runs the whole `TestSuite` and returns a `ValidationResult`.

1.1 DirectRS

Tree ensembles are among the most accurate models for tabular data, and among the hardest to explain. DirectRS post-processes a pre-trained ensemble (XGBoost, LightGBM, CatBoost) without retraining it. It extracts a global stretch matrix S' from the tree structure, then fits a per-tree Ridge regression on the stretched embedding $\varphi_t(\mathbf{x}) = [1,\, S'\mathbf{x}]$, replacing each tree's piecewise-constant leaf value with a piecewise-linear one. The result keeps the ensemble's accuracy while giving an *exact* additive decomposition of every prediction,

$$f(\mathbf{x}) = c_0 + \sum_{j=1}^{p} c_j(\mathbf{x}),$$

where $c_j(\mathbf{x})$ is the contribution of feature j and the identity holds to machine precision — no sampling and no approximation, unlike post-hoc attribution. DirectRS supplies six interpretability methods: a global stretch analysis, exact local decomposition, four global-importance modes, a main-vs-interaction variance split, geometric interaction traces and a top pairwise-interaction table.

Classes: `MoDirectRSRegressor`, `MoDirectRSClassifier`.

1.2 GBDT Leaf Kernel

The GBDT leaf kernel re-reads a fitted ensemble rather than reshaping it. Every gradient-boosted prediction can be written as a weighted average of training labels, scores or residuals under the ensemble's own leaf co-membership kernel: two inputs are close when the trees route them to the same leaves. This turns an opaque GBDT into a nearest-neighbor model over a learned geometry and exposes the evidence behind each prediction without a separate surrogate. The wrapper takes a pre-trained XGBoost model and reuses its leaves; it does not retrain. On that geometry it offers five interchangeable prediction heads — `exact_gbdt`, `gnw_label`, `gnw_leaf`, `krr` and `residual_gnw` — from an exact reconstruction of the base model to smoothed and repaired variants, with a `nominate_head` procedure that selects one on validation evidence. Every prediction carries a Top-k Leaf-Evidence (TKLE) ledger: the neighbors that supported it, the effective sample size behind it, the label and score dispersion among those neighbors, and a local calibration coefficient. A `predict_with_context` method swaps the training memory for a user-supplied pool while keeping the kernel fixed, which supports counterfactual and case-based audits without retraining.

Classes: `MoGBDTKernelRegressor`, `MoGBDTKernelClassifier`.

1.3 FuseKernel

A single kernel encodes one prior over functions. A tree co-membership kernel is sharp and piecewise constant, so it captures abrupt thresholds but oversmooths a gentle trend; an RBF kernel is the opposite; a learned spectral kernel captures periodic and high-order structure neither of the others sees. No one kernel is right for every dataset. FuseKernel forms a convex mixture of up to three channels — a tree co-membership kernel (XGBoost, LightGBM or CatBoost, at one depth or several fused), an RBF kernel and a learned multi-scale spectral kernel — and decodes the mixture with a single kernel ridge:

$$\hat{f}(\mathbf{x}) = \sum_c w_c \, K_c(\mathbf{x}, \mathcal{S}) \, (\mathbf{K}_{\mathcal{S}\mathcal{S}} + \lambda \mathbf{I})^{-1} \mathbf{y}_{\mathcal{S}}, \qquad w_c \geq 0, \; \sum_c w_c = 1.$$

A convex combination of positive semi-definite kernels is again a valid kernel, so the simplex of mixtures interpolates between the priors. Each channel is normalized to a unit diagonal, so the mixture stays unit-diagonal and the ridge λ is identifiable. The mixture weights and the ridge are chosen on a held-out query fold that no channel has trained on, which keeps the fusion from over-crediting a channel that memorizes its own training rows. FuseKernel exposes an inherent functional-ANOVA interpretation, driven by the spectral channel's learned per-feature relevance, and closed-form Gaussian-process prediction intervals for reliability diagnostics without a conformal wrapper.

Classes: `MoFuseKernelRegressor`, `MoFuseKernelClassifier`.

1.4 ICL-MoE

ICL-MoE (In-Context Learning Mixture of Experts) builds on a fitted DirectRS model. DirectRS maps each sample into the embedding $\mathbf{z} = S'\mathbf{x}$; two samples that land near each other in $\mathbf{z}$-space share similar split-routing patterns across all trees, so proximity there is structural similarity from the model's perspective rather than raw feature proximity. ICL-MoE replaces hard tree-based leaf routing with soft k-nearest-neighbor attention gating in that space: for each query it finds its contextual neighbors and fits a local expert on them, correcting the DirectRS leaf experts with a

kNN-weighted local Ridge on their residuals. This is the tabular analogue of in-context learning, where the context is the set of nearby training examples in the learned embedding.

Classes: `MoDirectRSICLRegressor`, `MoDirectRSICLClassifier`.

2 New validation diagnostics

Version 2.0 adds four diagnostics that locate where a model fails, not only how well it does on average. The first decomposes the aggregate error; the remaining three find and, in one case, repair the regions that drive it.

2.1 Error decomposition

A model with acceptable aggregate accuracy can fail in specific subpopulations. The error decomposition splits a single aggregate number into per-cluster contributions so the failure is visible.

For classification the global AUC decomposes exactly into intra-cluster and inter-cluster blocks,

$$\mathrm{AUC}_{\mathrm{global}} = \sum_{i=1}^{K} \sum_{j=1}^{K} w_{ij}\, \mathrm{AUC}_{ij},$$

where the diagonal terms AUC_{ii} measure discrimination *within* a cluster and the off-diagonal terms AUC_{ij} measure how the model ranks one cluster's positives against another's negatives. A low diagonal entry marks a subgroup the model cannot separate; a low off-diagonal entry marks cross-subgroup miscalibration, a fairness concern the global AUC hides.

For regression the within-cluster MSE decomposes into four interpretable terms,

$$\mathrm{MSE}_k = \underbrace{(\bar{f}_k - \bar{y}_k)^2}_{\text{calibration}} + \underbrace{\sigma^2_{f,k}}_{\text{prediction variance}} + \underbrace{\sigma^2_{y,k}}_{\text{outcome variance}} - \underbrace{2\,\mathrm{Cov}_k(f, y)}_{\text{alignment}},$$

which separates reducible error (poor calibration or misspent capacity) from the irreducible outcome variance $\sigma^2_{y,k}$ that no model can beat in that region. Summing over clusters adds a between-cluster resolution block that measures how well the clustering separates distinct outcome levels. Under binary outcomes with constant within-cluster predictions this recovers Murphy's Brier decomposition exactly (reliability, resolution, uncertainty), so the framework generalizes the classical result to continuous outcomes.

The clustering that drives the decomposition is itself error-aware. Error-Aware Random Forest (EARF) clustering trains a random forest to predict residual magnitude from the features and clusters samples by the forest's leaf co-occurrence, so samples are grouped by shared error structure rather than feature similarity. The clusters can span non-contiguous feature regions, precisely because error patterns cut across geometric boundaries.

2.2 Weakness detection and repair with FuseKernel

FuseKernel adds a way to find a model's weak regions and repair them. The detection step uses the fused kernel as a learned geometry: `diagnose_weak_clusters` embeds the fitted kernel with a Nyström spectral map, partitions the data into clusters in that geometry and reports the model metric per cluster on train and test. Breaking performance down per cluster surfaces the weak

regions without having to guess where to look; the result ranks the worst clusters and maps each test row to its cluster. Detection is cheap — only the tree co-membership kernel at the base model's depth is needed, so the RBF and spectral channels can stay off.

The repair step fits a residual-driven Mixture of Experts (`MoMoERegressor` / `MoMoEClassifier`). With `cluster_method="ltc"` it clusters samples by their learning trajectories — where the base model struggles — fits a specialized expert per cluster and routes between them with a gate. The experts can carry the same monotonicity constraints as the base model, so the repaired model stays interpretable. The gain is largest where the base model was weakest.

2.3 Weakness diagnostics on the GBDT leaf kernel

The GBDT leaf kernel carries its own weakness diagnostic. `diagnose_weakness` ranks the ensemble's behavioral clusters by their contribution to validation loss and, for the weakest cluster, identifies the features that distinguish it from the rest by Jensen–Shannon divergence between the in-cluster and out-of-cluster feature histograms. The features with the largest divergence name the weak cluster — they describe the kind of case the ensemble currently struggles with. This is residual weakness slicing anchored in the model's own leaf kernel rather than in raw feature axes.

2.4 AMIF weakness-region detection

`TestSuite.diagnose_weakness_region` locates weak regions of the input space with an Adversarial Mutual Information Forest (AMIF) pipeline: geometry scoring via an adversarial random forest, mutual-information scoring via a cross-validated random forest, then 2D binning with a configurable grid and a weak-fraction cutoff. Features are ranked by Jensen–Shannon divergence between the weak and non-weak regions. The outputs include region-performance heatmaps for train and test, divergence and mutual-information rankings, score distributions and per-feature distributions, and confusion matrices for classification.

3 Packaging and licensing

MoDeVa 2.0 changes how the package is distributed:

- **Compiled distribution.** The package ships as compiled extensions rather than editable source.
- **NumPy 1.x.** MoDeVa 2.0 requires `numpy<=1.26.4`. Install it into an environment with NumPy 1.x; it will not import under NumPy 2.x.
- **License tiers.** Access is governed by a signed license with tier quotas (DEMO, DEVELOPER, ENTERPRISE) that cap dataset rows, registered models and worker count. Dependency availability is checked at import.

4 Upgrading

Install into a fresh environment with NumPy 1.x and a license valid for your tier. The 1.x public API is unchanged, so existing `DataSet`, `ModelZoo`, `TestSuite` and `Pipeline` code runs without modification. The new models and diagnostics are additive.

Where to read more

The monograph covers each feature with its theory, an executed notebook and worked labs:

- DirectRS — Chapter 14, *DirectRS: Geometric Post-Processing of Tree Ensembles*.
- GBDT leaf kernel and its weakness diagnostics — Chapter 15, *The GBDT Leaf Kernel: Generalized Nadaraya–Watson Heads*.
- FuseKernel — Chapter 16, *FuseKernel: Fusing Tree, RBF, and Learned Spectral Kernels*.
- ICL-MoE — Chapter 17, *ICL-MoE: In-Context Learning Mixture of Experts*.
- Error decomposition (AUC and MSE) and EARF clustering — Chapter 21, *Accuracy Diagnostics*.
- Weakness detection, weak-cluster repair and AMIF — Chapter 22, *Residual Analysis and Weakness Detection*.