TABLELOOP: TABULARIZING MULTIMODAL DATA VIA RUBRIC EVOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many prediction tasks depend on information distributed across text, images, and structured measurements. We introduce TableLoop, a flexible framework that connects language and vision-language models with tabular predictors through learned natural-language rubrics, closing the loop between what a model measures and how useful those measurements are for downstream prediction. A frozen scoring model applies each rubric to produce explicit numeric features, while a tabular predictor's validation performance and errors guide iterative revisions to the rubric. The resulting interpretable features are combined with structured measurements for final prediction. TableLoop is modular: it requires no language-model tuning and supports interchangeable scoring models, tabular predictors, and evolution methods. Across nine tasks spanning clinical risk, persuasion outcomes, essay quality, peer-review decisions, skin-lesion malignancy, and fungal species identity, TableLoop outperforms both structured-data-only prediction and calibrated direct prediction by a larger language or vision-language model in 26 of 27 evaluated configurations. Notably, TableLoop using a smaller open-weight scorer (Qwen3-32B/Qwen3-VL-32B) can outperform direct prediction with a frontier model (Claude Sonnet 4.5), showing that a smaller model can support stronger predictions when used to extract task-specific features rather than predict the target directly. Beyond prediction, TableLoop produces reusable, interpretable feature tables grounded in real observations, offering a potential source of training and adaptation data for tabular foundation models and an inspectable interface for multimodal-to-table learning.

1 INTRODUCTION

Many prediction problems require combining unstructured data, such as natural language and images, with structured measurements (Shi et al., 2021; Zhang et al., 2020; Tang et al., 2024). For example, predicting a patient's medical outcome may require interpreting doctors' notes alongside vital signs and laboratory results (Wang et al., 2025). These sources can provide complementary evidence, but using them together requires a model to both interpret unstructured inputs and learn how they relate to the recorded measurements. Language and vision language models offer a way to extract information from text and images, while tabular predictors learn relationships among explicit features. Connecting these capabilities offers a practical route to prediction from heterogeneous data.

One approach is to convert unstructured inputs into numeric features that a tabular predictor can combine with existing measurements (Bisercic et al., 2023; Balek et al., 2025). Natural-language rubrics provide an explicit interface for this conversion: a rubric defines properties to assess and instructions for scoring them, and a language or vision-language model assigns a score to each property for every input (Hashemi et al., 2024; Lam et al., 2024). This makes the extracted information interpretable: researchers can inspect both what each feature is intended to measure and the scores assigned to individual inputs. The tabular representation also makes it straightforward to combine these features with metadata or add further measurements and features as they become available. However, a plausible rubric may not be the most useful one for prediction. Its criteria may be difficult to score consistently, redundant, or weakly related to the outcome of interest. Feature selection can identify useful features within an existing rubric (Guyon & Elisseeff, 2003), but it cannot introduce properties that the rubric does not measure or improve how they are defined and scored. Prior work has explored learning concepts and natural-language features from labeled examples and prediction

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

errors (Ludan et al., 2024; Zhong et al., 2024; Feng et al., 2025; Jojic & Jojic, 2025), including approaches that construct rubric-based representations, optimize feature-generation instructions, or refine grading rubrics (Demirel et al., 2026; Cosma et al., 2026; Chu et al., 2026). Our goal is to optimize a complete rubric—its criteria and scoring instructions—using feedback from a predictor trained on the resulting scores.

We introduce **TableLoop**, a framework for combining structured and unstructured data through rubrics learned from predictive feedback (Figure 1). A language or vision language model applies each candidate rubric to produce numeric features, a tabular predictor learns from these features, and its performance and errors on validation data guide a rubric-evolution model in revising the scoring criteria and instructions. TableLoop supports interchangeable scoring models, tabular predictors, and rubric-evolution methods (Khattab et al., 2024; Guo et al., 2024; Agrawal et al., 2025). We also introduce a novel method for rubric optimization: successive halving with complementary merge (SH-merge), which repeatedly combines candidate rubrics and evaluates the resulting smaller population on progressively more data.

The output of TableLoop is not just a predictor. Each learned rubric defines an explicit procedure for converting unstructured observations into compatible tabular features, and the resulting tables can be inspected, stored, and reused with different predictive models. Because rubric criteria are expressed in natural language, the framework can be applied across domains without requiring a manually specified feature set for each task. This also creates an opportunity to construct tabular datasets from existing labeled text and image collections, potentially broadening the data available for training and adapting tabular foundation models.

We evaluate TableLoop on nine prediction tasks spanning clinical risk, persuasion outcomes, essay quality, peer-review decisions, skin-lesion malignancy, and fungal species identity. With TabPFN as the tabular predictor (Hollmann et al., 2025), TableLoop outperforms both a predictor using structured measurements alone and a larger language or vision language model making predictions directly from all available inputs across all nine tasks we evaluated (Figure 2). These results show that a smaller language model coupled with learned rubrics and a tabular predictor can outperform prediction by a larger model while retaining explicit intermediate features for inspection, reuse, and interpretability.

Our contributions are threefold:

- **A flexible framework for combining structured and unstructured data.** TableLoop connects language and vision language models with tabular predictors through reusable, inspectable feature tables and supports interchangeable models and rubric-evolution methods.
- **Learning what to measure through feedback.** We optimize rubric criteria and scoring instructions using validation performance. We also introduce SH-merge, which combines candidate rubrics while progressively increasing their evaluation budgets.
- **Evaluation across tasks and design choices.** Across nine tasks we evaluate, TableLoop exceeds both metadata-only and direct-prediction baselines in each task. Ablations examine rubric initialization, evolution method, ensemble selection, structured-data integration, and predictor choice.

2 Related Work

Combining structured and unstructured data. Prior work combines text and structured measurements through multimodal models or by converting structured records into text for language models (Shi et al., 2021; Tang et al., 2024; Hegselmann et al., 2023). An alternative is to convert unstructured inputs into explicit features for a tabular predictor. Language models can generate features by transforming existing columns (Hollmann et al., 2023b; Han et al., 2024; Abhyankar et al., 2026) or extracting attributes from text (Bisercic et al., 2023; Balek et al., 2025; Wang et al., 2025), while tabular foundation models provide powerful predictors for the resulting tables (Hollmann et al., 2023a; 2025; Google Research, 2026). Rubrics define these features through named criteria and scoring instructions, supporting inspection and concept exploration (Lam et al., 2024; Koh et al., 2020). LLM-Rubric learns to combine scores from a fixed rubric (Hashemi et al., 2024), while other work constructs a rubric from labeled examples (Demirel et al., 2026). TableLoop uses predictive feedback to revise the rubric itself, then combines its scores with structured measurements.

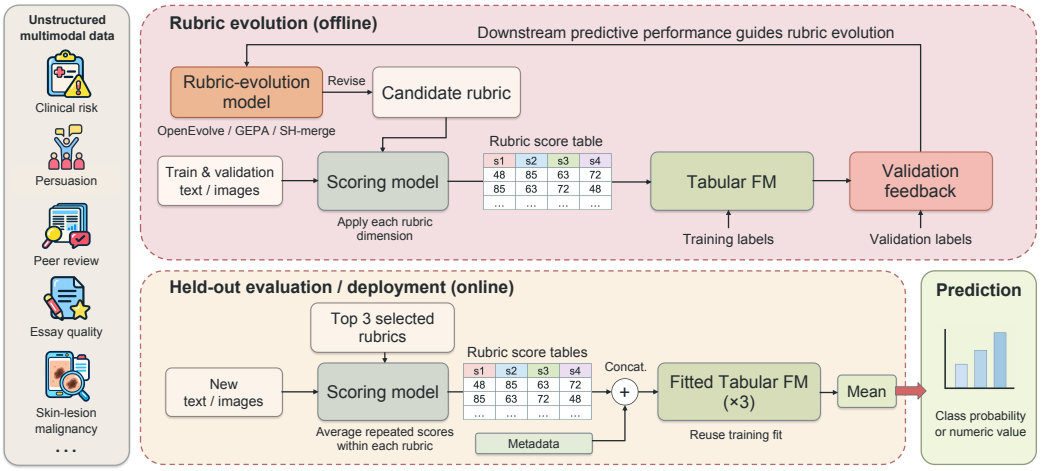


Figure 1: **Overview of TableLoop. Top:** A language or vision-language model applies candidate rubrics to produce numeric features. A tabular predictor's validation performance and errors guide rubric revisions, without updating model weights. **Bottom:** The three rubrics selected by rubric-only validation performance produce feature tables that are combined with structured measurements. A separate tabular predictor is fitted for each rubric, and their predictions are averaged.

Learning features from predictive feedback. Several methods use labeled examples and prediction errors to learn concepts or natural-language features (Ludan et al., 2024; Zhong et al., 2024; Feng et al., 2025; Jojic & Jojic, 2025; Zheng et al., 2026). TableLoop builds on this direction by optimizing a complete rubric, allowing its criteria and scoring instructions to change together. Text Bottleneck Models incrementally propose and refine concepts using misclassified examples, retaining those that improve a linear predictor (Ludan et al., 2024). FEST builds a bank of semantic and executable features from contrasts between positive and negative examples and uses decision-tree importance to guide which features remain in the bank (Khurana et al., 2026). Cosma et al. (2026) take a different approach, optimizing a prompt that tells a language model how to generate features. TableLoop instead optimizes the rubric that defines the features and how to score them. We compare these optimization targets in an appendix ablation, where direct rubric evolution performs better on both evaluated tasks (Appendix D). Other work revises grading rubrics to improve an LLM's assigned grades (Chu et al., 2026), whereas TableLoop optimizes rubric scores as inputs to a separate predictor. Combining learned concepts with structured information also has precedent: BC-LLM combines note-derived concepts with a tabular model's risk score (Feng et al., 2025). TableLoop supplies the original structured measurements alongside rubric scores to the final predictor.

Optimizing rubrics efficiently. Prompt-optimization methods provide tools for proposing and revising instructions using performance feedback (Yang et al., 2024; Guo et al., 2024; Fernando et al., 2024; Khattab et al., 2024; Agrawal et al., 2025). TableLoop applies these methods to rubric search, where evaluating each candidate requires scoring observations and fitting a tabular predictor. Our SH-merge method draws on successive halving (Jamieson & Talwalkar, 2016): the candidate population shrinks across rounds while each candidate is evaluated on more data. Before reducing the population, it merges higher- and lower-ranked rubrics, allowing criteria from weaker candidates to contribute to subsequent rounds.

3 Methods

3.1 Framework Overview

TableLoop combines unstructured inputs, such as text or images, with structured measurements by learning task-specific scoring rubrics (Figure 1). Each rubric defines a set of named properties and instructions for assigning numeric scores to them. A *scoring model*, implemented by a language or vision language model, applies the rubric to each unstructured input, producing one feature per rubric

dimension. To reduce variation between responses, we score each input M times and average the scores separately for each dimension. The scoring model receives neither the instance's structured measurements nor its prediction target.

We divide the data into training, validation, and held-out test sets. Starting from an initial rubric, TableLoop alternates between evaluating candidate rubrics and proposing revisions. For each candidate, we apply a tabular foundation model to its rubric scores and training targets, then evaluate validation performance using the task-specific metric: ROC-AUC or PR-AUC for classification, and MAE for regression. The *rubric-evolution model* uses validation performance and high-error examples to propose changes to the rubric's criteria and scoring instructions. The search method determines which candidates and feedback are used to generate revisions, as described below. Structured measurements are excluded throughout rubric evaluation and selection, and the test set is not used during search. Language or vision-language model weights remain fixed; optimization occurs through rubric revisions and fitting the tabular predictors.

After the search budget is exhausted, we select the three rubrics with the best rubric-only validation performance. For each selected rubric, we combine its scores with the structured measurements and apply the tabular foundation model to the test set (with only the training set as context). At test time, we average the three predictors' probabilities for classification or scalar predictions for regression:

$$\widehat{y}_i = \frac{1}{K} \sum_{k=1}^{K} f_k \left([s_i^{R_k}; m_i] \right), \qquad K = 3, \tag{1}$$

where $s_i^{R_k}$ contains the scores assigned to instance i under rubric R_k, m_i contains its structured measurements, and f_k is the corresponding final predictor. The brackets denote feature concatenation. TableLoop supports interchangeable scoring models, tabular predictors, and rubric-evolution methods.

3.2 Rubric Evolution Methods

We implement TableLoop with OpenEvolve, GEPA, and our successive halving with complementary merge method (SH-merge). OpenEvolve maintains a population of candidate rubrics across multiple groups, or islands, together with an archive of strong candidates (Sharma, 2025). At each iteration, the rubric-evolution model rewrites a selected parent using its validation performance, high-error examples, and search history. The revised rubric is evaluated and added to the population. Our GEPA implementation instead emphasizes detailed feedback on a small validation minibatch (Agrawal et al., 2025). The rubric-evolution model receives the selected rubric together with each example's input, target, rubric scores, and tabular prediction. A proposed revision is first evaluated on the same minibatch. Promising revisions receive full-validation evaluation and enter the candidate pool. Both methods run to fixed search budgets, with implementation settings given in the experimental protocol.

SH-merge combines parallel exploration with repeated rubric merging. We first evaluate the initial rubric, then ask the rubric-evolution model to generate C variants using sampled high-error validation examples. Within each round, all candidates are evaluated on the same sampled training and validation instances and ranked by validation performance. We pair the highest-ranked candidate with the lowest-ranked, the second-highest with the second-lowest, and so on. For each pair, the rubric-evolution model receives both rubrics, their performance and validation errors, and SHAP feature importance (Lundberg & Lee, 2017), then produces one child rubric by combining or revising their criteria. Thus, every parent can contribute to the next round, while the population halves. As the population shrinks, the number of instances used to evaluate each candidate doubles:

$$B_r = \min\left(2^{r-1}B,\ N_{\text{train}} + N_{\text{val}}\right), \tag{2}$$

where B is the initial per-candidate sample budget, r indexes evaluation rounds, and N_{train} and N_{val} are the split sizes. Training and validation instances retain their separate roles throughout. This schedule keeps the scorer-instance budget approximately constant per round while evaluating later candidates on more data. Evaluations can run in parallel within each round, and the search ends after a fixed number of rounds. The final round uses the full training and validation sets, and its three best rubrics form the final ensemble.

4 EXPERIMENTAL SETUP

Datasets. We evaluate nine prediction tasks drawn from six datasets. Four tasks predict ICLR paper acceptance within separate research areas, using paper text and structured paper attributes. The training set uses ICLR 2025 papers, while validation and testing use ICLR 2026 papers. The remaining tasks predict essay proficiency from ELLIPSE corpus essays and writing-related measurements (Crossley et al., 2023), donation outcomes from Persuasion For Good dialogues and participant personal/psychological attributes (Wang et al., 2019), invasive ventilation within 24 hours of a radiology report from MIMIC-IV clinical text and measurements (Johnson et al., 2023c;a;b), skin cancer from PAD-UFES-20 lesion images and patient and lesion attributes (Pacheco et al., 2020), and fungal species identity from a set of five visually similar Russula species based on photographs and observation records in the Danish Fungi dataset (Picek et al., 2022). We refer to the MIMIC-IV ventilation task as MIMIC-MV (Mechanical Ventilation) and Persuasion For Good as Persuasion below. We evaluate MAE for ELLIPSE, PR-AUC for MIMIC-MV, and ROC-AUC for the remaining tasks. The appendix describes the inputs, splits, and evaluation procedures.

TableLoop. We use Claude Sonnet 4.5 for rubric evolution, except on the MIMIC-MV task, where we use locally hosted Qwen3-235B (the data use agreement requires running only local language models). Inputs are scored with Qwen3-32B for text and Qwen3-VL-32B-Instruct for images (Yang et al., 2025; Bai et al., 2025), averaging three responses per instance. TabPFN is the predictor in all main experiments. All searches begin with a single-dimension rubric. OpenEvolve runs for 50 iterations, GEPA for 30 full-evaluation iterations, and SH-merge for five evaluation rounds starting from 64 candidates. Rubric evolution and selection exclude structured measurements; these are added when fitting the final three-predictor ensemble. No language or vision-language model weights are updated.

Baselines. We compare TableLoop with metadata-only TabPFN and direct language or vision-language model prediction on every task, using the same data splits. Metadata-only TabPFN predicts from the structured measurements alone. Direct prediction asks Claude Sonnet 4.5 to predict the target from the task description together with the structured and unstructured inputs, and uses locally hosted Qwen3-235B on the MIMIC-MV task. Its zero-shot predictions are then calibrated on the validation set.

Related feature engineering method. FEST (Feature Engineering with Self-Evolving Trees) (Khurana et al., 2026), designed for binary classification, provides a closely related comparison on our seven binary tasks: like TableLoop, it converts unstructured inputs into interpretable features for downstream prediction. Specifically, FEST discovers semantic predicates and executable features from contrasting positive and negative examples, consolidates them into a shared feature bank, and iteratively prunes the bank using decision-tree importance. Because the authors do not release code, we reimplement FEST as closely as possible using the procedure, prompts, and hyperparameters specified in the paper. For a matched evaluation, we combine its selected features with the same structured measurements and use the same TabPFN predictor and data splits as TableLoop.

5 RESULTS

TableLoop outperforms both structured-measurement-only TabPFN and direct prediction by a larger language or vision language model in 26 of 27 configurations across nine tasks and three evolution methods (Figure 2). All three evolution methods exceed both baselines on eight tasks. On the MIMIC-MV prediction task, OpenEvolve and GEPA exceed both baselines, while SH-merge falls below metadata-only TabPFN. The best-performing TableLoop variant exceeds both baselines on every task.

The largest gains occur in ICLR acceptance prediction, where the best TableLoop variant improves ROC-AUC by 28.1–39.1% relative to the stronger of the two baselines. Across the remaining tasks, the corresponding improvements are 7.5% in MIMIC-MV PR-AUC, 1.7% in Persuasion ROC-AUC, 0.7% in PAD-UFES-20 lesion-level ROC-AUC, and 17.1% in Danish Fungi observation-level macro ROC-AUC, with a 1.4% reduction in ELLIPSE MAE. These gains span text and image inputs, tasks ranging from binary and multiclass classification to regression, and the temporal shift from ICLR 2025 training papers to ICLR 2026 evaluation papers.

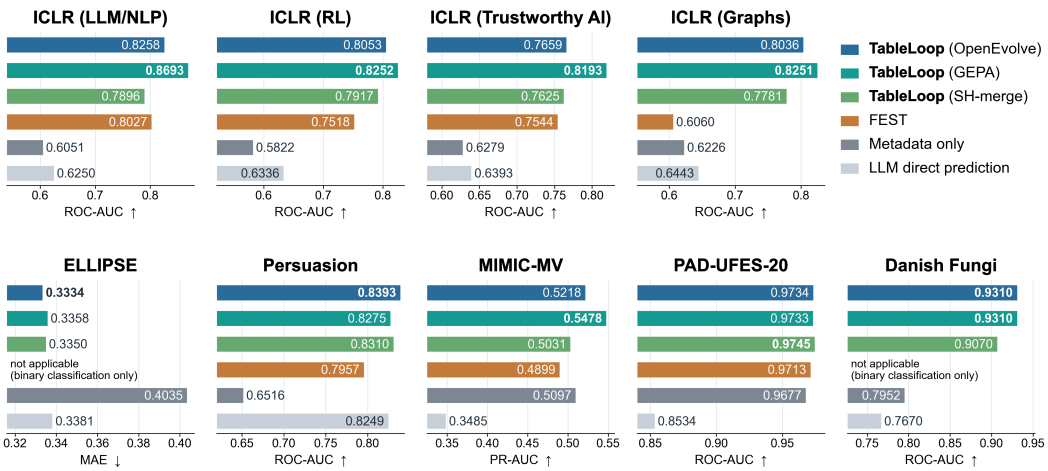


Figure 2: **TableLoop gains persist across evolution methods.** Results are shown for OpenEvolve, GEPA, and SH-merge using the top-three ensemble selected by rubric-only validation performance. Metadata is added after selection to fit the final TabPFN predictors. The baselines are metadata-only TabPFN and metadata-aware zero-shot direct prediction with validation-fitted affine calibration. For the seven binary classification tasks, it also reports results from FEST (Khurana et al., 2026). Each panel reports the native held-out test metric: MAE for ELLIPSE, PR-AUC for MIMIC-MV, and ROC-AUC otherwise.

TableLoop also consistently outperforms FEST across the seven binary classification tasks. OpenEvolve and GEPA outperform FEST on all seven tasks, while SH-merge does so on six, falling short only on ICLR LLM/NLP. The advantage is particularly pronounced on ICLR Graphs, where the relative improvements are 28.4–36.2%. These results support jointly evolving a complete rubric rather than constructing and pruning a bank of individually proposed features.

No single evolution method performs best on every task. GEPA leads on MIMIC-MV and all four ICLR categories, OpenEvolve on Persuasion and ELLIPSE, and SH-merge on PAD-UFES-20. The consistent gains across these methods support TableLoop as a flexible approach to combining unstructured information with structured measurements. They also show that a smaller model used to extract rubric features can support better predictions than direct use of a larger model.

Beyond the choice of evolution method, we test whether TableLoop is robust to the downstream tabular predictor by replacing TabPFN with TabFM (Google Research, 2026) and XGBoost (Chen & Guestrin, 2016). TabFM performs similarly to TabPFN, and both generally outperform XGBoost (Appendix Table 4), suggesting that the benefit of rubric-derived representations is not specific to TabPFN and that tabular foundation models are well suited as downstream predictors in TableLoop.

5.1 Evolution Improves Test Performance

To determine whether held-out performance improves as rubric evolution progresses, we conduct a retrospective truncation analysis of OpenEvolve. In the main experiments, the test set is evaluated only once after evolution is complete. For this ablation, we instead truncate the run after each iteration k, select the current top-three ensemble using rubric-only validation performance, add metadata when applying the corresponding TabPFN predictors, and evaluate that ensemble on the test set. For ICLR, the results are averaged across three folds. While the main experiments do not use early stopping, with each OpenEvolve run using a fixed 50-iteration budget, we also retrospectively consider validation-based early stopping, where patience p means that evolution stops after p consecutive iterations without an improvement in the best-so-far validation score.

Figure 3 shows that held-out performance generally improves throughout evolution across all nine tasks. The gains accumulate over multiple iterations as stronger rubrics enter the validation-selected top-three ensemble, although individual trajectories include plateaus and occasional reversals. The final ensembles achieve relative improvements of up to 14.5%, showing that improvements selected

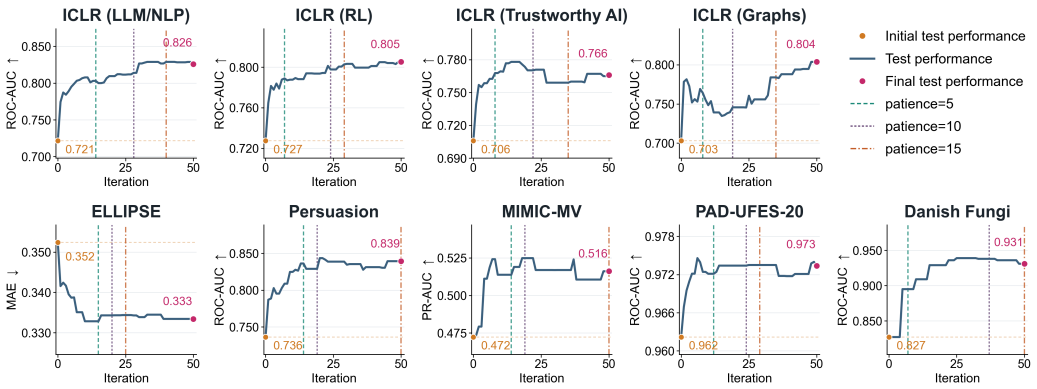


Figure 3: **Evolution improves held-out test performance.** At each iteration k of a 50-iteration OpenEvolve run, the current validation-selected top-three ensemble is evaluated on the test set. Iteration 0 is the naive initial rubric, and the dotted horizontal line marks its test performance. Vertical lines mark hypothetical early-stopping points under validation patience 5, 10, and 15.

on the validation set generally carry over to held-out test data. As a retrospective analysis, the vertical lines in Figure 3 mark where the validation-based early-stopping rule would have halted. If we introduced this form of early stopping, it would often reach a substantial fraction of the test gain at a smaller evaluation cost, but it can also halt too soon and miss later improvements.

The main experiments initialize evolution from a deliberately minimal, single-dimension rubric. To examine whether a more informative starting point improves the search, we additionally compare this naive initialization with expert-designed, task-specific rubrics for ICLR and ELLIPSE datasets. The naive initialization performs better across all five evaluated tasks, suggesting that starting from a broad rubric leaves greater flexibility for evolution to discover useful task-specific dimensions. Full details and results are provided in Appendix C.

5.2 Instance-Level Validation Feedback

TableLoop uses two forms of validation feedback during evolution: an aggregate validation metric and examples on which the current predictor makes large errors. To isolate the contribution of the latter, we rerun OpenEvolve while withholding the validation examples, labels, rubric scores, and predictions associated with these errors. The evolution model still receives the aggregate validation metric; all other settings remain unchanged.

Table 1: **Instance-level validation feedback improves rubric evolution.** OpenEvolve results with the standard feedback and with only the aggregate validation metric. Results use the validation-selected top-three ensemble aggregation. ICLR results are averaged across three folds.

Task	Metric	Instance feedback	Without Instance feedback
ICLR (LLM/NLP)	ROC-AUC ↑	0.8258	0.8055
ICLR (RL)	ROC-AUC ↑	0.8053	0.8044
ICLR (Trustworthy AI)	ROC-AUC ↑	0.7659	0.7318
ICLR (Graphs)	ROC-AUC ↑	0.8036	0.7653
ELLIPSE	MAE ↓	0.3334	0.3347
Persuasion	ROC-AUC ↑	0.8393	0.7747
MIMIC-MV	PR-AUC ↑	0.5218	0.2589
PAD-UFES-20	ROC-AUC ↑	0.9734	0.9686
Danish Fungi	ROC-AUC ↑	0.9310	0.9179

Withholding instance-level error feedback generally degrades held-out performance, even though the evolution model still observes the aggregate validation metric. The metric indicates whether a rubric performs well overall, but does not identify which distinctions the rubric fails to capture or score correctly. These results show that instance-level validation feedback is important for directing useful rubric revisions.

5.3 Does Evolution Discover Complementary Rubrics?

We examine whether rubric evolution expands the set of examples that can be predicted correctly and discovers rubrics with complementary strengths. To distinguish this from simply improving predictions on examples already handled well, we measure *coverage*: the fraction of validation instances for which at least one rubric-specific predictor produces a correct prediction. For classification, correctness uses a probability threshold of 0.5; for ELLIPSE, it requires an absolute error of at most ϵ.

Using rubric-only validation predictions from 50 OpenEvolve iterations, we compare four selections: the initial rubric (Ini.), the evolved rubric with the highest individual coverage (Cov-1), the three rubrics with the highest combined coverage (Cov-3), and the standard three rubrics selected by validation task performance (Top-3). Ini. versus Cov-1 tests whether evolution finds individual rubrics that correctly predict more examples, while comparing Cov-1 with the three-rubric selections tests whether different rubrics succeed on different examples.

Coverage increases from Ini. to Cov-1 on every task (Table 2), showing that evolution discovers individual rubrics that correctly predict a larger fraction of validation examples. Top-3 further exceeds Cov-1 on every task, indicating that the selected rubrics have complementary successes rather than predicting the same examples correctly. For example, on ICLR LLM/NLP, coverage rises from 58.7% for Ini. to 68.4% for Cov-1 and 84.0% for Top-3 before adding metadata.

We also test whether explicitly selecting for coverage improves the final ensemble. Cov-3 achieves the highest validation coverage by construction, but its averaged predictions yield worse held-out performance than Top-3 on six of nine tasks. Having at least one correct predictor for an example does not guarantee that averaging the predictors will improve the final prediction. These results support using coverage to assess complementary strengths while retaining validation task performance as the default rubric-selection criterion.

Table 2: **Evolution discovers rubrics with complementary predictive strengths.** Validation coverage is the fraction of examples predicted correctly by at least one selected rubric-specific predictor, using rubric scores alone. Ini. uses the initial rubric; Cov-1 selects the individual rubric with highest coverage; Cov-3 selects three rubrics by their combined coverage; and Top-3 selects three by validation task performance. The final two columns report held-out performance after adding structured measurements and averaging predictions. ICLR results are averaged across three folds.

		Val. coverage				Test metric	
Task	Metric	Ini.	Cov-1	Cov-3	Top-3	Cov-3	Top-3
ICLR (LLM/NLP)	ROC-AUC ↑	0.5865	0.6837	0.8646	0.8404	0.8187	0.8258
ICLR (RL)	ROC-AUC ↑	0.6189	0.6831	0.8811	0.8207	0.7907	0.8053
ICLR (Trustworthy AI)	ROC-AUC ↑	0.5597	0.6511	0.8573	0.8083	0.7789	0.7659
ICLR (Graphs)	ROC-AUC ↑	0.6903	0.7308	0.9365	0.8693	0.7960	0.8036
ELLIPSE ($\epsilon = .5$)	MAE ↓	0.7045	0.7521	0.8473	0.8304	0.3352	0.3334
Persuasion	ROC-AUC ↑	0.6483	0.7517	0.8690	0.8138	0.8208	0.8393
MIMIC-MV	PR-AUC ↑	0.8527	0.8549	0.8637	0.8615	0.5252	0.5161
PAD-UFES-20	ROC-AUC ↑	0.6736	0.8012	0.9258	0.8754	0.9740	0.9734
Danish Fungi	ROC-AUC ↑	0.3980	0.7245	0.8980	0.8061	0.9208	0.9353

5.4 Complementarity with Metadata

For some tasks, the available metadata is comprehensive: it includes attributes closely related to the unstructured multimodal data and achieves strong predictive performance on its own. To examine the contribution of TableLoop when this overlap is reduced, we conduct a controlled ablation on

PAD-UFES-20. We remove nine potentially overlapping lesion-specific measurements and symptoms: region, two diameter measurements, elevation, itching, growth, pain, change, and bleeding, while retaining 12 patient-level demographic, background, and risk-history fields. The unstructured input, selected rubrics, and TabPFN configuration are held fixed; only the metadata set supplied to TableLoop and its matched metadata-only baseline changes.

Table 3: Controlled metadata ablation on PAD-UFES-20, measured by test lesion-level ROC-AUC. The reduced setting removes nine lesion-specific fields while retaining 12 patient-level demographic, background, and risk-history fields.

Metadata set	Fields	Metadata only	TableLoop	TableLoop gain
Full	21	0.9677	0.9734	$+0.0057$
Patient-level only	12	0.8709	0.9345	$+0.0636$

The gain from TableLoop over metadata-only prediction increases from 0.0057 with full metadata to 0.0636 after the potentially overlapping fields are removed. This larger gap shows that TableLoop extracts useful predictive information from the unstructured multimodal data and that its rubric features complement the remaining structured metadata.

6 Discussion

TableLoop provides a flexible framework for combining structured measurements with information from text and images. By learning what a language or vision-language model should measure, it turns unstructured inputs into explicit features that tabular predictors can use alongside existing data. Predictive feedback guides revisions to the rubric's criteria and scoring instructions, allowing the representation to adapt to the task without finetuning the language model.

Across nine tasks, TableLoop outperforms both structured-data-only learning and calibrated direct prediction by a larger language or vision-language model, showing that a smaller model used to extract task-specific features can support stronger predictions than direct use of a larger model. The ablations further show that evolution can improve held-out performance and discover rubrics with complementary predictive strengths, while the resulting features remain useful across different tabular predictors. Although the learned rubrics are inspectable, their alignment with the intended concepts warrants further human evaluation. More broadly, as with other validation-guided optimization methods, studies across additional datasets, data splits, and scorer LLMs would help assess robustness.

Efficiency is an important consideration because rubric evolution requires repeatedly scoring examples and evaluating candidate representations, with costs depending on the dataset size, input length, and search budget. This optimization takes place offline: once the rubrics are selected, predictions on new inputs require only the fixed scoring procedure and downstream tabular predictors. Using a smaller scoring model therefore supports efficient deployment, with the initial search cost amortized over subsequent predictions. Initial experiments on ELLIPSE show that subsampling training instances during evolution, while retaining the full validation set, can reduce scoring requirements while maintaining competitive predictive performance (see appendix). More efficient candidate evaluation, adaptive search budgets, and reduced scoring repetition are promising future directions.

Another direction for future work is to compare rubric-derived features with embeddings extracted from language or vision-language models and supplied to a tabular foundation model. Such a comparison would help assess the value of explicitly task-adapted features relative to general-purpose representations. Embeddings could also be combined with rubric scores and structured measurements to test whether they capture complementary informationand to assess the tradeoff between compact, inspectable features and richer latent representations.

Beyond its final predictions, TableLoop produces reusable feature tables grounded in real observations. A learned rubric can be applied to additional inputs, and its scores can be stored, inspected, and used with other predictors. This creates opportunities to convert labeled text and image collections from different domains into tabular datasets, potentially expanding the data available for training and adapting tabular foundation models. Although this application remains to be evaluated, it highlights the broader utility of the framework: learning explicit, task-relevant representations that make unstructured information accessible to tabular learning.

AI USE STATEMENT

We used generative AI tools to edit the manuscript for readability and to aid retrieval and discovery of related work. We did not use generative AI tools for any research tasks requiring disclosure under the ICLR AI policy. The language models used as scorers and evolution engines are part of the proposed method and experimental pipeline, not authoring assistants. We have reviewed all AI-assisted work: authors checked edited text against the paper's methods and results, and verified suggested citations against the original sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We access MIMIC-IV, MIMIC-IV-ED, and MIMIC-IV-Note under their data-use requirements. These deidentified clinical records remain in our local computing environment, and all language-model processing of them uses locally hosted models; we do not send them to external model providers or redistribute them. The other datasets contain potentially sensitive human-generated text, demographic attributes, or clinical images. We use only the released data for the stated research purposes, and make no attempt to identify participants.

REPRODUCIBILITY STATEMENT

The main text describes the TableLoop pipeline, rubric evolution and selection, baselines, and evaluation metrics. Appendix H provides additional details on dataset construction and splits, input features, models, hyperparameters, search budgets, missing-value handling, and baseline configurations. The appendix also reports extended baseline and ablation results. MIMIC data must be obtained independently through its authorized access process; the other datasets can be obtained from their original sources.

REFERENCES

Nikhil Abhyankar, Parshin Shojaee, and Chandan K. Reddy. LLM-FE: Automated feature engineering for tabular data with LLMs as evolutionary optimizers. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=qvI35hkpOO.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025. URL https://arxiv.org/abs/2507.19457.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Vojtěch Balek, Lukáš Sýkora, Vilém Sklenák, and Tomáš Kliegr. LLM-based feature generation from text for interpretable machine learning. *Machine Learning*, 114:241, 2025. doi: 10.1007/s10994-025-06867-1. URL https://link.springer.com/article/10.1007/s10994-025-06867-1.

Aleksa Bisercic, Mladen Nikolic, Mihaela van der Schaar, Boris Delibasic, Pietro Lio, and Andrija Petrovic. Interpretable medical diagnostics with structured data extraction by large language models. *arXiv preprint arXiv:2306.05052*, 2023. URL https://arxiv.org/abs/2306.05052.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Yucheng Chu, Hang Li, Kaiqi Yang, Yasemin Copur-Gencturk, Joseph Krajcik, Namsoo Shin, and Jiliang Tang. Confusion-aware rubric optimization for LLM-based automated

grading. In *Proceedings of the International Conference on Educational Data Mining*, 2026. URL https://educationaldatamining.org/edm2026/proceedings/2026.EDM.full-papers.232/index.html.

Adrian Cosma, Oleg Szehr, David Kletz, Alessandro Antonucci, and Olivier Pelletier. Automatic prompt optimization for dataset-level feature discovery. *arXiv preprint arXiv:2601.13922*, 2026. URL https://arxiv.org/abs/2601.13922.

Scott A. Crossley, Yu Tian, Perpetual Baffour, Alex Franklin, Youngmeen Kim, Wesley Morris, Meg Benner, Aigner Picou, and Ulrich Boser. The English Language Learner Insight, Proficiency and Skills Evaluation (ELLIPSE) Corpus. *International Journal of Learner Corpus Research*, 9(2): 248–269, 2023. doi: 10.1075/ijlcr.22026.cro. URL https://doi.org/10.1075/ijlcr.22026.cro.

Ilker Demirel, Lawrence Shi, Zeshan Hussain, and David Sontag. Llms can construct powerful representations and streamline sample-efficient supervised learning. *arXiv preprint arXiv:2603.11679*, 2026. URL https://arxiv.org/abs/2603.11679.

Jean Feng, Avni Kothari, Lucas Zier, Chandan Singh, and Yan Shuo Tan. Bayesian concept bottleneck models with LLM priors. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=oXSkzIXgbk.

Chrisantha Fernando, Dylan Sunil Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 13481–13544, 2024. URL https://proceedings.mlr.press/v235/fernando24a.html.

Rita González-Márquez and Dmitry Kobak. Learning representations of learning representations. In *Data-centric Machine Learning Research (DMLR) Workshop at ICLR*, 2024. URL https://arxiv.org/abs/2404.08403.

Google Research. Introducing: TabFM: A zero-shot foundation model for tabular data. Google Research release and model card, 2026. URL https://research.google/blog/introducing-tabfm-a-zero-shot-foundation-model-for-tabular-data/.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZG3RaNIsO8.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003. URL https://www.jmlr.org/papers/v3/guyon03a.html.

Sungwon Han, Jinsung Yoon, Sercan O. Arik, and Tomas Pfister. Large Language Models Can Automatically Engineer Features for Few-Shot Tabular Learning. *arXiv preprint arXiv:2404.09491*, 2024. URL https://arxiv.org/abs/2404.09491.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. Llm-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13806–13834, 2024. doi: 10.18653/v1/2024.acl-long.745. URL https://aclanthology.org/2024.acl-long.745/.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 5549–5581, 2023. URL https://proceedings.mlr.press/v206/hegselmann23a.html.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations*, 2023a. URL https://arxiv.org/abs/2207.01848.

Noah Hollmann, Samuel Müller, and Frank Hutter. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c2df4c35cdbee764ebb9e9d0acd5197-Abstract-Conference.html.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 2025. doi: 10.1038/s41586-024-08328-6. URL https://doi.org/10.1038/s41586-024-08328-6.

Kevin Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 240–248, 2016. URL https://proceedings.mlr.press/v51/jamieson16.html.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Leo Anthony Celi, Roger Mark, and Steven Horng. MIMIC-IV-ED. *PhysioNet*, 2023a. doi: 10.13026/5ntk-km72. URL https://doi.org/10.13026/5ntk-km72. Version 2.2.

Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV-Note: Deidentified free-text clinical notes. *PhysioNet*, 2023b. doi: 10.13026/1n74-ne17. URL https://doi.org/10.13026/1n74-ne17. Version 2.2.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10:1, 2023c. doi: 10.1038/s41597-022-01899-x. URL https://doi.org/10.1038/s41597-022-01899-x.

Marko Jojic and Nebojsa Jojic. GenZ: Foundational models as latent variable generators within traditional statistical models. *arXiv preprint arXiv:2512.24834*, 2025. URL https://arxiv.org/abs/2512.24834.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.03714.

Varun Khurana, Vijval Ekbote, Vashu Chauhan, Yaman K. Singla, Rajiv Ratn Shah, and Balaji Krishnamurthy. Bridging expert knowledge and automated feature engineering via self-evolution. *arXiv preprint arXiv:2606.08800*, 2026. URL https://arxiv.org/abs/2606.08800.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348, 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Michelle S. Lam, Janice Teoh, James A. Landay, Jeffrey Heer, and Michael S. Bernstein. Concept induction: Analyzing unstructured text with high-level concepts using LLooM. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024. doi: 10.1145/3613904.3642830. URL https://doi.org/10.1145/3613904.3642830.

Josh Magnus Ludan, Qing Lyu, Yue Yang, Liam Dugan, Mark Yatskar, and Chris Callison-Burch. Interpretable-by-design text understanding with iteratively generated concept bottleneck. *arXiv preprint arXiv:2310.19660*, 2024. URL https://arxiv.org/abs/2310.19660. Version 2; first posted in 2023.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1705.07874.

Andre G. C. Pacheco, Gustavo R. Lima, Amanda S. Salomão, Breno Krohling, Igor P. Biral, Gabriel G. de Angelo, Fábio C. R. Alves, Jr., José G. M. Esgario, Alana C. Simora, Pedro B. C. Castro, Felipe B. Rodrigues, Patricia H. L. Frasson, Renato A. Krohling, Helder Knidel, Maria C. S. Santos, Rachel B. do Espírito Santo, Telma L. S. G. Macedo, Tania R. P. Canuto, and Luíz F. S. de Barros. PAD-UFES-20: A skin lesion dataset composed of patient data and clinical images collected from smartphones. *Data in Brief*, 32:106221, 2020. doi: 10.1016/j.dib.2020.106221. URL https://doi.org/10.1016/j.dib.2020.106221.

Lukáš Picek, Milan Šulc, Jiří Matas, Thomas S. Jeppesen, Jacob Heilmann-Clausen, Thomas Læssøe, and Tobias Frøslev. Danish Fungi 2020 - Not Just Another Image Recognition Dataset. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1525–1535, 2022. doi: 10.1109/WACV51458.2022.00334. URL https://doi.org/10.1109/WACV51458.2022.00334.

Asankhaya Sharma. OpenEvolve: An open-source evolutionary coding agent. https://github.com/algorithmicsuperintelligence/openevolve, 2025. Open-source implementation inspired by AlphaEvolve.

Xingjian Shi, Jonas Mueller, Nick Erickson, Mu Li, and Alexander J. Smola. Benchmarking multimodal AutoML for tabular data with text fields. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/9bf31c7ff062936a96d3c8bd1f8f2ff3-Abstract-round2.html.

Zhiqiang Tang, Haoyang Fang, Su Zhou, Taojiannan Yang, Zihan Zhong, Cuixiong Hu, Katrin Kirchhoff, and George Karypis. AutoGluon-Multimodal (AutoMM): Supercharging multimodal AutoML with foundation models. In *Proceedings of the Third International Conference on Automated Machine Learning*, volume 256 of *Proceedings of Machine Learning Research*, pp. 15/1–35. PMLR, 2024. URL https://proceedings.mlr.press/v256/tang24a.html.

Jiayi Wang, Jacqueline Jil Vallon, Nikhil V. Kotha, Neil Panjwani, Xi Ling, Margaret Redfield, Sushmita Vij, Sandy Srinivas, John Leppert, Mark K. Buyyounouski, and Mohsen Bayati. Scaling clinician-grade feature generation from clinical notes with multi-agent language models. *arXiv preprint arXiv:2508.01956*, 2025. URL https://arxiv.org/abs/2508.01956.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5635–5649, 2019. doi: 10.18653/v1/P19-1566. URL https://aclanthology.org/P19-1566/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.03409.

Dongdong Zhang, Changchang Yin, Jucheng Zeng, Xiaohui Yuan, and Ping Zhang. Combining structured and unstructured data for predictive models: a deep learning approach. *BMC Medical Informatics and Decision Making*, 20:280, 2020. doi: 10.1186/s12911-020-01297-6. URL https://link.springer.com/article/10.1186/s12911-020-01297-6.

Lifang Zheng, Hanmo Liu, and Kani Chen. Adaptive concept discovery for interpretable few-shot text classification. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/61f425da6e0a201b8fe1454601abfba5-Abstract-Conference.html.

Ruiqi Zhong, Heng Wang, Dan Klein, and Jacob Steinhardt. Explaining datasets in words: Statistical models with natural language parameters. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2520. URL https://papers.nips.cc/paper_files/paper/2024/hash/90c4537a301e9545bb4c60219f2992b1-Abstract-Conference.html.

A Downstream Predictor Results

Although the main experiments use TabPFN, TableLoop is not tied to a specific downstream tabular predictor. We compare TabPFN with TabFM and XGBoost while holding the selected top-three ensemble rubrics and input features fixed for each evolution method. Table 4 reports the results across OpenEvolve, GEPA, and SH-merge.

Table 4: **Complete evolution-method-by-backbone results.** Each dataset uses its native metric. OpenEvolve uses a 50-iteration budget; GEPA uses 30 full-evaluation iterations. "Direct prediction" denotes the metadata-aware zero-shot direct prediction score; "Metadata-only" uses only metadata; "TableLoop" denotes the default method with its validation-selected top-three ensemble and metadata.

Dataset	Setting	OpenEvolve			GEPA			SH-merge		
		PFN	TabFM	XGB	PFN	TabFM	XGB	PFN	TabFM	XGB
ICLR (LLM/NLP) ROC-AUC ↑	Direct prediction					0.6250				
	Metadata-only	0.6051	0.6150	0.5476	0.6051	0.6150	0.5476	0.6051	0.6150	0.5476
	TableLoop	0.8258	0.8301	0.7980	0.8693	**0.8721**	0.8493	0.7896	0.7914	0.7697
ICLR (RL) ROC-AUC ↑	Direct prediction					0.6336				
	Metadata-only	0.5822	0.5787	0.5510	0.5822	0.5787	0.5510	0.5822	0.5787	0.5510
	TableLoop	0.8053	0.8059	0.7804	0.8252	**0.8257**	0.7925	0.7917	0.7919	0.7818
ICLR (Trustworthy AI) ROC-AUC ↑	Direct prediction					0.6393				
	Metadata-only	0.6279	0.6327	0.5512	0.6279	0.6327	0.5512	0.6279	0.6327	0.5512
	TableLoop	0.7659	0.7625	0.7394	0.8193	**0.8196**	0.8042	0.7625	0.7586	0.7420
ICLR (Graphs) ROC-AUC ↑	Direct prediction					0.6443				
	Metadata-only	0.6226	0.5917	0.5367	0.6226	0.5917	0.5367	0.6226	0.5917	0.5367
	TableLoop	0.8036	0.7936	0.8049	**0.8244**	0.8238	0.8039	0.7781	0.7681	0.7636
ELLIPSE MAE ↓	Direct prediction					0.3381				
	Metadata-only	0.4035	0.4034	0.4227	0.4035	0.4034	0.4227	0.4035	0.4034	0.4227
	TableLoop	0.3334	**0.3315**	0.3427	0.3358	0.3345	0.3431	0.3350	0.3330	0.3417
Persuasion ROC-AUC ↑	Direct prediction					0.8249				
	Metadata-only	0.6516	0.6342	0.5519	0.6516	0.6342	0.5519	0.6516	0.6342	0.5519
	TableLoop	**0.8393**	0.8301	0.8113	0.8275	0.8328	0.8088	0.8310	0.8229	0.8133
MIMIC-MV PR-AUC ↑	Direct prediction					0.3485				
	Metadata-only	0.5097	0.4964	0.4219	0.5097	0.4964	0.4219	0.5097	0.4964	0.4219
	TableLoop	0.5218	0.5151	0.4646	**0.5478**	0.5255	0.4538	0.5031	0.4930	0.4591
PAD-UFES-20 ROC-AUC ↑	Direct prediction					0.8534				
	Metadata-only	0.9677	0.9693	0.9577	0.9677	0.9693	0.9577	0.9677	0.9693	0.9577
	TableLoop	0.9734	0.9750	0.9706	0.9733	0.9758	0.9740	0.9745	**0.9772**	0.9737
Danish Fungi ROC-AUC ↑	Direct prediction					0.7670				
	Metadata-only	0.7952	0.8009	0.8103	0.7952	0.8009	0.8103	0.7952	0.8009	0.8103
	TableLoop	0.9307	**0.9373**	0.9322	0.9307	0.9372	0.9327	0.9074	0.9157	0.9125

B Direct LM/VLM Prediction Results

We sweep shot count, affine calibration, and the presence of metadata to choose a strong direct-prediction baseline for comparison with TableLoop. Scores are aggregated by aligning instances across three seeds, averaging the per-instance score, and computing the metric once. An affine map is fitted on validation predictions and applied unchanged to test. Several zero-shot runs are deterministic and therefore use a single seed. Tables 6–8 report the full grid.

Few-shot prompting substantially improves uncalibrated predictions on ELLIPSE, but these gains largely disappear after validation-fitted affine calibration. For example, with metadata, Claude Sonnet 4.5's raw MAE decreases from 0.6105 with zero shots to 0.4671 with ten shots, whereas calibrated zero-shot prediction achieves a lower MAE than any tested few-shot setting (0.3381 versus 0.3500–0.3528). The same pattern holds for Qwen3-32B, while Claude without metadata shows only small, inconsistent gains from demonstrations after calibration. This suggests that much of the benefit of demonstrations on ELLIPSE concerns matching the target score scale, which calibration can address directly. Across classification tasks, few-shot examples sometimes improve performance, but their

effects vary by task and model. We therefore use calibrated zero-shot prediction as a consistent direct-prediction baseline, while reporting the few-shot comparisons here. These findings also motivate exploring ways to use labeled data beyond including examples in the prediction prompt. TableLoop uses predictive feedback to revise what the scoring model measures and trains a downstream predictor to combine the resulting features with structured measurements.

Table 5: ELLIPSE direct prediction scores (MAE ↓).

		Without metadata		With metadata	
Model	Shot	Raw	Affine	Raw	Affine
Claude Sonnet 4.5	0	0.6316	0.3431	0.6105	0.3381
	3	0.4679	0.3428	0.5017	0.3528
	5	0.4566	0.3413	0.4885	0.3524
	10	0.4648	0.3480	0.4671	0.3500
Qwen3-32B	0	0.5983	0.3928	0.5727	0.3831
	3	0.3985	0.4045	0.4166	0.4090
	5	0.3952	0.4042	0.4130	0.4089
	10	0.3986	0.4053	0.4067	0.4107

Table 6: Persuasion direct prediction scores (ROC-AUC ↑).

		Without metadata		With metadata	
Model	Shot	Raw	Affine	Raw	Affine
Claude Sonnet 4.5	0	0.7923	0.7923	0.8249	0.8249
	3	0.7268	0.7268	0.7692	0.7692
	5	0.8129	0.8129	0.8019	0.8018
	10	0.7912	0.7908	0.8033	0.8033
Qwen3-32B	0	0.7610	0.7610	0.7667	0.7667
	3	0.7807	0.7804	0.7572	0.7572
	5	0.7952	0.7952	0.7628	0.7632
	10	0.7652	0.7652	0.7593	0.7594

Table 7: MIMIC-MV direct prediction scores (PR-AUC ↑). Affine maps are monotone here, so ranking metrics are unchanged except for occasional clipping or ties.

		Without metadata		With metadata	
Model	Shot	Raw	Affine	Raw	Affine
Qwen3-235B	0	0.2721	0.2721	0.3485	0.3485
	3	0.2908	0.2908	0.3307	0.3307
	5	0.3110	0.3110	0.3400	0.3399
	10	0.3063	0.3063	0.3315	0.3315
Qwen3-32B	0	0.2465	0.2465	0.3001	0.3001
	3	0.2613	0.2613	0.3128	0.3128
	5	0.2809	0.2809	0.3244	0.3244
	10	0.3032	0.3032	0.3168	0.3168

Table 8: PAD-UFES-20 direct prediction score (lesion-level ROC-AUC ↑). The selected Claude Sonnet 4.5 zero-shot setting includes the image and metadata.

Model	Shot	Raw	Affine
Claude Sonnet 4.5	0	0.8537	0.8534

Table 9: ICLR direct prediction without metadata (ROC-AUC ↑). Each cell is the mean of three fold-level metrics; within a fold, scores are averaged across seeds before the metric is computed.

Category	Model	Shot	Raw	Affine
LLM/NLP	Claude Sonnet 4.5	0	0.6532	0.6532
		3	0.6473	0.6478
		5	0.6336	0.6335
		10	0.6668	0.6669
	Qwen3-32B	0	0.6447	0.6447
		3	0.6236	0.6236
		5	0.6365	0.6365
		10	0.6320	0.6320
RL	Claude Sonnet 4.5	0	0.6622	0.6622
		3	0.6640	0.6641
		5	0.6517	0.6514
		10	0.6718	0.6717
	Qwen3-32B	0	0.6955	0.6955
		3	0.6825	0.6826
		5	0.6714	0.6709
		10	0.6864	0.6865
Trustworthy AI	Claude Sonnet 4.5	0	0.6680	0.6680
		3	0.6433	0.6442
		5	0.6520	0.6521
		10	0.6664	0.6661
	Qwen3-32B	0	0.6129	0.6129
		3	0.5796	0.5795
		5	0.6012	0.6011
		10	0.5901	0.5901
Graphs	Claude Sonnet 4.5	0	0.6803	0.6803
		3	0.7184	0.7178
		5	0.6378	0.6383
		10	0.6303	0.6297
	Qwen3-32B	0	0.5761	0.5761
		3	0.5576	0.5576
		5	0.6314	0.6314
		10	0.6106	0.6106

Table 10: ICLR metadata-aware Claude Sonnet 4.5 zero-shot direct prediction (ROC-AUC ↑), the setting used in the main results. Scores are means of three fold-level metrics.

Category	Raw	Affine
LLM/NLP	0.6250	0.6250
RL	0.6336	0.6336
Trustworthy AI	0.6393	0.6393
Graphs	0.6443	0.6443

Across these grids, few-shot demonstrations do not consistently improve over shot 0, affine calibration is essential for ELLIPSE MAE while leaving ranking metrics essentially unchanged, and providing metadata typically helps. The metadata-aware, affine-calibrated zero-shot setting is therefore the strongest overall protocol, and we use it as the direct-prediction baseline in the main experiments when comparing against TableLoop.

B.1 Adding a Direct Prediction Score

We finally investigate whether a direct prediction score from a frontier model can provide complementary signal and further improve TableLoop. We consider two integration strategies: *feature concatenation*, which supplies the calibrated direct score to TabPFN alongside the rubric features and

metadata, and *score fusion*, which fits a validation-trained ridge model over the TableLoop and direct predictions.

Table 11: Direct-prediction-score augmentation for the 50-iteration OpenEvolve top-three ensemble with TabPFN and metadata. Direct scores are produced by Claude Sonnet 4.5, except on MIMIC-MV, where privacy restrictions require Qwen3-235B. ICLR results are averaged over three folds.

Task	Metric	TableLoop	Feature concat.	Score fusion
ICLR (LLM/NLP)	ROC-AUC ↑	0.8258	**0.8316**	0.8292
ICLR (RL)	ROC-AUC ↑	0.8053	**0.8108**	0.8056
ICLR (Trustworthy AI)	ROC-AUC ↑	0.7659	**0.7848**	0.7781
ICLR (Graphs)	ROC-AUC ↑	0.8036	**0.8207**	0.8070
ELLIPSE	MAE ↓	0.3334	**0.3243**	0.3254
Persuasion	ROC-AUC ↑	**0.8393**	0.8385	0.8351
MIMIC-MV	PR-AUC ↑	0.5161	**0.5338**	0.5234
PAD-UFES-20	ROC-AUC ↑	**0.9734**	0.9719	0.9720
Danish Fungi	ROC-AUC ↑	**0.9308**	0.9261	0.9190

Overall, the direct prediction score can further improve TableLoop, yielding higher held-out performance on six of the nine tasks. Among the two integration strategies, feature concatenation outperforms score fusion on seven tasks, although often by a small margin. This pattern favors incorporating the direct score at the feature level, where TabPFN can learn its relationship with the rubric features and metadata, over combining the final predictions post hoc.

C Naive versus Expert Rubric Initialization

The outcome of prompt evolution may depend on the rubric used to initialize the search. We therefore compare two initializations for TableLoop with OpenEvolve. The *naive initialization* contains only one broad, target-related dimension: overall paper quality for ICLR and overall writing proficiency for ELLIPSE. The *expert initialization* starts from multiple task-specific dimensions. For ICLR, these dimensions are adapted from the official review criteria, including soundness, presentation, significance, originality, overall recommendation, and reviewer confidence. For ELLIPSE, they follow its human analytic scoring rubric and cover cohesion, syntax, vocabulary, phraseology, grammar, and conventions (Crossley et al., 2023). We keep all other settings fixed and run both variants for 50 iterations. We restrict this comparison to ICLR and ELLIPSE because both have established, externally defined evaluation criteria: the ICLR review criteria and the ELLIPSE human scoring rubric. These provide a basis for expert initialization without requiring us to design a task-specific expert rubric, whose quality could otherwise confound the comparison.

Table 12: **Effect of rubric initialization on TableLoop with OpenEvolve.** Both variants use the same fixed 50-iteration budget; ICLR results are averaged over three folds.

Initialization	ICLR (LLM/NLP)	ICLR (RL)	ICLR (Trustworthy AI)	ICLR (Graphs)	ELLIPSE
	AUC ↑				MAE ↓
Naive	**0.8258**	**0.8053**	**0.7659**	**0.8036**	**0.3334**
Expert	0.7848	0.7693	0.7379	0.7800	0.3349

As shown in Table 12, naive initialization performs better on ELLIPSE and all four ICLR categories. This suggests that a detailed initial rubric is not necessarily a better starting point for evolution. A predefined decomposition may restrict the directions explored during optimization, whereas a single broad dimension leaves OpenEvolve more flexibility to discover and reorganize dimensions based on validation feedback. This supports our use of naive initialization in the main experiments.

D Direct Rubric Evolution versus Instruction Evolution

We test whether predictive feedback is more useful for revising a rubric directly or for revising instructions that generate a rubric. TableLoop takes the first approach: the text being evolved defines the features and how to score them. For comparison, we implement an *instruction evolution* variant inspired by Cosma et al. (2026), in which OpenEvolve instead revises a prompt describing how a rubric should be constructed. A separate language-model call then uses that prompt to generate the rubric.

Both approaches use 50 OpenEvolve iterations and the same scoring and TabPFN evaluation pipeline. In the instruction-evolution variant, each rubric-generation call also receives 16 validation examples selected using the parent candidate's predictions: the four best-predicted and four worst-predicted examples from each class. The generated rubric is scored and evaluated, and its validation performance guides subsequent revisions to the generation instructions. Neither approach uses test examples or labels during optimization.

Table 13: **Direct rubric evolution outperforms instruction evolution on both tested tasks.** Both approaches use 50 OpenEvolve iterations and a three-rubric ensemble selected by rubric-only validation performance. Structured measurements are added when fitting the final TabPFN predictors. Higher values are better.

Task	Metric	TableLoop	Instruction evolution
Persuasion	ROC-AUC ↑	**0.8393**	0.8316
MIMIC-MV	PR-AUC ↑	**0.5218**	0.4947

Direct rubric evolution achieves higher held-out performance on both tasks, improving ROC-AUC by 0.0077 on Persuasion and PR-AUC by 0.0271 on MIMIC-MV (Table 13). These results support revising the scoring criteria and instructions directly. One possible explanation is that direct evolution lets predictive feedback guide changes to the exact rubric being evaluated, whereas instruction evolution must influence that rubric through an additional generation step. The comparison supports this design choice on the two tested tasks, but does not establish the cause of the performance difference.

E Reducing Evolution Cost through Training Subsampling

Evaluating a candidate rubric requires applying the scoring model to both training and validation examples, making scoring the dominant cost of evolution. We test whether this cost can be reduced by subsampling the training set while retaining the full validation set for candidate evaluation, selection, and error feedback. We conduct this ablation on ELLIPSE using the same naive initialization and 50-iteration OpenEvolve budget as the main experiment. For the subsampled runs, a new training subset is drawn for each candidate evaluation, while validation feedback is sampled from pooled high- and low-error examples. Final top-three ensemble ensembles are evaluated using the full training and test sets.

Table 14: **Training subsampling preserves ELLIPSE performance.** Held-out test MAE for OpenEvolve using different fractions of the training set during evolution. Validation always uses the full split. Lower values are better.

Training fraction	Rubric features	+ structured measurements
10%	0.3463	0.3323
30%	0.3453	0.3348
50%	0.3517	0.3387
100%	0.3450	0.3334

Across the tested fractions, training subsampling maintains competitive test performance while reducing the number of examples scored during evolution. The 10% run even slightly improves MAE

after structured measurements are added. Training subsampling therefore provides a practical way to reduce evolution cost, although aggressive subsampling may be less stable on datasets with fewer training examples.

Table 15: **Fixed and resampled training subsets give similar results.** ELLIPSE test MAE with a 30% training subset during evolution. Lower values are better.

Sampling strategy	Rubric features	+ structured measurements
Fixed subset	**0.3430**	**0.3337**
Resampled subset	0.3459	0.3344

Fixed and resampled training subsets produce similar results, suggesting that the choice between these two sampling strategies has little effect in this experiment.

F Feature Importance Feedback

We test whether adding global mean-absolute SHAP importance for each rubric dimension to OpenEvolve validation feedback improves Persuasion. The comparison uses the 50-iteration OpenEvolve top-three ensemble with TabPFN.

Table 16: Persuasion feature-importance feedback (50-iteration OpenEvolve, TabPFN top-three ensemble, ROC-AUC ↑).

Evaluation	Control	+ FI feedback	Difference
Rubric features only	0.8135	0.8100	−0.0035
+ metadata	0.8393	0.8455	+0.0062

The effect is small and changes sign once metadata is added, so we do not include SHAP importance in the default evolution feedback.

G LLM-Generated Confidence Features

For MIMIC-MV, we additionally ask Qwen3-235B to report a confidence score, calibrate it on validation data, and add it to the rubric features. OpenEvolve uses the 50-iteration top-three ensemble; GEPA uses the top-three ensemble from the first 30 iterations.

Table 17: MIMIC-MV confidence-feature experiment with TabPFN and the top-three ensemble (OpenEvolve 50 iterations, GEPA 30 iterations).

Evolution method	Added features	Shot	ROC-AUC	PR-AUC
OpenEvolve	Confidence	0	0.8043	0.3382
	Confidence	10	0.8069	0.3464
	Metadata + confidence	0	0.8427	0.5314
	Metadata + confidence	10	0.8464	0.5344
	Metadata + confidence + direct score	0	0.8472	0.5121
	Metadata + confidence + direct score	10	0.8472	0.5312
GEPA	Confidence	0	0.8035	0.3648
	Confidence	10	0.8102	0.3816
	Metadata + confidence	0	0.8485	0.5485
	Metadata + confidence	10	0.8533	0.5590
	Metadata + confidence + direct score	0	0.8579	0.5712
	Metadata + confidence + direct score	10	0.8538	0.5597

Confidence yields mixed gains: they are modest without metadata, larger in some metadata settings, and not monotone in shot count when the direct prediction score is also included. Because the pattern is unstable across evolution methods and feature packages, we do not include confidence in the default TableLoop representation.

H Experimental Setup Details

H.1 Datasets

ICLR papers. Using the topic labels provided by the ICLR dataset (González-Márquez & Kobak, 2024), we define four acceptance-prediction tasks: Graphs, Reinforcement Learning, Trustworthy AI, and LLM/NLP. Each task has its own evolved rubrics. Within each category, all ICLR 2025 papers form the training set, while ICLR 2026 papers are divided into validation and test sets under three stratified folds. The unstructured input consists of the title, abstract, and body text parsed with Docling, truncated at 12,000 scoring-model tokens. Structured features comprise the numbers of authors, figures, and tables, and appendix length. The target is acceptance versus rejection. We compute test ROC-AUC separately for each fold and report the mean across the three folds.

ELLIPSE. ELLIPSE contains essays written by English-language learners, with human-assigned Overall proficiency scores from 1.0 to 5.0 in increments of 0.5 (Crossley et al., 2023). We predict this score using the writing prompt and essay text alongside nine structured features describing length, lexical diversity, readability, paragraph structure, and prompt type. We partition the released training data into training and validation sets and retain the official test set for held-out evaluation. Performance is measured by mean absolute error (MAE).

Persuasion For Good. Persuasion For Good contains dialogues in which one participant attempts to persuade another to donate part of their task payment to Save the Children (Wang et al., 2019). We predict whether the persuadee makes an actual donation using the full dialogue transcript and structured features describing dialogue statistics and participant demographics, personality, moral foundations, and values. All dialogues involving the same persuadee are assigned to the same split. Performance is measured by ROC-AUC.

MIMIC-MV. We construct this dataset from MIMIC-IV, MIMIC-IV-ED, and MIMIC-IV-Note (Johnson et al., 2023c;a;b). The unstructured input contains the emergency-department chief complaint and radiology report. Structured features include demographics, triage measurements, the last recorded ED vital signs, and laboratory values. Taking the radiology-report time as the prediction time, we predict whether invasive ventilation begins within the following 24 hours. All positive ventilation events occur after the report time. To reduce textual leakage, we additionally exclude instances whose reports explicitly mention an endotracheal tube, intubation, tracheostomy, or mechanical ventilation. Because positive outcomes are uncommon, performance is measured by PR-AUC.

PAD-UFES-20. PAD-UFES-20 contains 2,298 smartphone clinical images of 1,891 lesions from 1,373 patients (Pacheco et al., 2020). We predict cancer—basal cell carcinoma, squamous cell carcinoma, or melanoma—versus actinic keratosis, nevus, or seborrheic keratosis. Inputs comprise a lesion image and 21 structured variables describing demographics, lesion characteristics, symptoms, risk factors, and medical history. We use a deterministic 70/15/15 training/validation/test split, grouping instances that share a patient, lesion, or byte-identical image to prevent overlap across splits. For lesions with multiple images, we average prediction scores before computing lesion-level ROC-AUC.

Danish Fungi. We use field photographs and observation records from Danish Fungi 2020 (Picek et al., 2022). The task is to identify one of five visually similar Russula species: *Russula emetica*, *Russula paludosa*, *Russula rosea*, *Russula nobilis*, and *Russula velenovskyi*. The subset contains 1,135 images from 654 observations. Inputs comprise a photograph and three structured variables describing habitat, substrate, and observation month. We use a deterministic 70/15/15 training/validation/test split, stratified by species and assigning every image from the same observation to the same split. For observations with multiple images, we average prediction scores before computing observation-level macro ROC-AUC.

H.2 TABLELOOP CONFIGURATION

Models. For rubric evolution, we use Claude Sonnet 4.5 at temperature 0.7 on all tasks except MIMIC-MV. Claude Sonnet 4.5 was released before the ICLR 2026 acceptance decisions were announced, reducing the risk of exposure to those decisions during model training. For MIMIC-MV, we use locally hosted Qwen3-235B to keep clinical data within our local environment under the dataset's data-use restrictions. For scoring, we use Qwen3-32B on text tasks (Yang et al., 2025) and Qwen3-VL-32B-Instruct on image tasks, both at temperature 0.1. Each instance is scored three times, and responses are averaged separately for each rubric dimension. All main experiments use TabPFN with default estimator settings and random seed 42. No language-model weights are updated.

Shared evolution settings. All methods start from a naive rubric containing one dimension. Candidate evaluation and validation-based rubric selection use rubric scores alone. Structured measurements are added only after selection, when fitting the final tabular predictors. We reject candidates if fewer than $95\%$of their combined training and validation scoring responses parse successfully. Missing rubric values are imputed using training-set medians.

OpenEvolve. We use a population size of 50, an archive size of 5, and three islands, with migration every two iterations. Feedback includes the parent's validation performance and the five highest-error validation instances, ranked by confident mistakes on classification tasks and by absolute residual on regression.å Each run stops after 50 iterations.

GEPA. We use metric-aligned epsilon-greedy selection with $\epsilon = 0.3$, a reflection minibatch of 10 validation instances, and no merge operator. Each run stops after 30 full-evaluation iterations, giving a scoring-model call budget comparable to 50 OpenEvolve iterations.

SH-merge. We begin with $C = 64$ candidates and use $D = 5$ evaluation rounds. The initial per-candidate sample budget is

$$B = \left\lceil \frac{N_{\text{train}} + N_{\text{val}}}{2^{D-1}} \right\rceil, \tag{3}$$

where N_{train} and N_{val} denote the split sizes. With five rounds, the initial budget is one sixteenth of the combined training and validation set size, rounded up. The budget doubles between rounds, capped at the full dataset, while the candidate population halves through merging. The five rounds therefore evaluate 64, 32, 16, 8, and 4 candidates, respectively. The final round uses all training and validation instances in their respective roles, and the three best rubrics are selected from its four candidates.

We ran each experiment, varying the downstream tabular predictor and evolution method once, with a consistent random seed.

H.3 BASELINES

Metadata-only. This baseline applies TabPFN to the structured features alone. It uses the same data splits and TabPFN configuration as TableLoop, isolating the predictive contribution of the rubric features.

Direct prediction. The direct-prediction baseline receives the task description, unstructured input, and structured features, then outputs a probability for classification or a scalar prediction for regression. We use Claude Sonnet 4.5 for ICLR, ELLIPSE, Persuasion For Good, PAD-UFES-20, and Danish Fungi. For MIMIC-MV, we use locally deployed Qwen3-235B to keep the clinical data within our local environment. All calls use zero-shot prompting at temperature 0. We fit an affine calibration map $\widetilde{y} = as + b$ to validation predictions and targets, then apply it unchanged to test predictions. Reported performance is computed from the calibrated test predictions.

FEST. FEST (Feature Engineering with Self-Evolving Trees) (Khurana et al., 2026) discovers interpretable features from contrasting positive and negative examples through two complementary streams. Its semantic features are natural-language predicates whose presence is assessed by a language model, while its deterministic features are implemented as executable code. At each

iteration, FEST consolidates semantically redundant candidates, adds the resulting features to a shared bank, trains a decision tree on their encodings, and removes features with consistently low importance. We reimplement this discovery-mode procedure without expert-provided feature seeds, following the algorithmic structure and prompts described in the original paper. The original implementation uses GPT-4o-mini for its language-model stages and Qwen3-Embedding-4B for semantic consolidation; we use Claude Sonnet 4.5 for feature generation and consolidation, Qwen3-32B for semantic-feature encoding, Qwen3-VL-32B-Instruct for image inputs, and the same Qwen3-Embedding-4B model. For MIMIC-MV, all language-model stages use locally deployed Qwen models. To match the TableLoop evaluation protocol, we concatenate the selected FEST features with the same structured features and fit the same TabPFN predictor, rather than the classifier suite used in the original paper. We evaluate FEST on the seven binary classification tasks using the same training, validation, and test splits as TableLoop.