MAPLE: MULTI-AGENT PRIOR LEARNING FOR CONSTRUCTING TREE ENSEMBLES

Tuan-Kiet Nguyen Viet[1], **Ba-Thinh Nguyen**[2], **Thanh-Trung Huynh**[3†], **Huy-Hieu Pham**[4†]
[1]Hanoi University of Science and Technology [2]Vietnam National University Hanoi
[3]VinUniversity [4]VinUni-Illinois Smart Health Center, VinUniversity
[†]Coressponding authors: {trung.ht,hieu.ph}@vinuni.edu.vn

ABSTRACT

Tree ensembles based on bagging and boosting remain highly effective for tabular data, yet their construction typically relies on uniform or heuristic feature sampling strategies that overlook task-specific prior knowledge. We introduce MAPLE, a framework for Multi-Agent Prior Learning that integrates learned feature priors directly into the process of constructing tree ensembles. By leveraging multiple sources of inductive bias, MAPLE enables the ensemble to incorporate informative priors while preserving diversity. Experiments on multiple tabular benchmarks demonstrate that MAPLE consistently improves predictive performance and robustness over standard tree ensembles and prior-agnostic baselines, while remaining scalable and computationally efficient.

Code is available at https://github.com/HaiAu2501/MAPLE.

1 INTRODUCTION

Tabular data is a central modality in many real-world machine learning applications, including finance, healthcare, and scientific decision-making (Borisov et al., 2024). Although deep learning models dominate a wide range of other domains such as computer vision and natural language processing, decision tree-based methods remain highly effective for tabular learning (Grinsztajn et al., 2022; Shwartz-Ziv & Armon, 2022). Their simplicity, strong empirical performance, and inherent interpretability make them particularly well-suited for structured data.

More recently, the success of large language models (LLMs) has stimulated new research directions in tabular learning. With capabilities such as semantic reasoning, in-context learning, and broad domain knowledge, LLMs offer significant potential for integration into tabular pipelines, complementing classical machine learning models rather than replacing them. Below, we summarize several representative research themes and works along this direction.

Tabular Prediction. Several studies explore the use of LLMs as direct predictors for tabular data by serializing structured inputs into natural language prompts. TabLLM (Hegselmann et al., 2023) applies in-context learning to perform tabular prediction tasks using LLMs, demonstrating the feasibility of prompt-based tabular inference.

Automated Feature Generation and Selection. Recent work primarily uses LLMs for automated feature engineering. Several approaches focus on *feature generation*, synthesizing feature transformations from dataset or task context, including CAAFE (Hollmann et al., 2023), FeatLLM (Han et al., 2024), OCTree (Nam et al., 2024), LFG (Zhang et al., 2024), and LLM-FE (Abhyankar et al., 2025), while LLM-Select (Jeong et al., 2024) studies LLM-based *feature selection*.

Tree Induction and Rule Refinement. LLMs have also been incorporated into the learning and refinement of decision trees. Prior work explores LLM-assisted decision logic and rule induction (Liu et al., 2025; Knauer et al., 2025), and DeLTa (Ye et al., 2025) refines decision tree rules using LLM-generated corrections, enabling post-hoc improvement of tree-based predictions.

Tabular Data Preprocessing. Another line of research applies LLMs to tabular data preprocessing. Curated LLM (Seedat et al., 2024) employs LLMs for data curation and noise filtering, while LLM-Forest (He et al., 2025) integrates LLM reasoning into forest models for data imputation in the presence of missing values.

Taken together, these studies substantially enhance the capabilities of tree-based models by integrating LLM-derived knowledge at different stages of the learning pipeline. However, several important aspects remain underexplored. First, *scalability is a key concern*, as many existing techniques focus on improving a single decision tree, while tree ensembles are widely known to achieve significantly stronger performance in practice. Second, *the role of multi-agent settings has received limited attention*, particularly regarding how diverse perspectives from multiple LLMs can be coordinated and exploited effectively. Finally, when extending from individual trees to ensembles, a new problem emerges: *how to integrate LLMs into ensemble construction, selection, and coordination*.

Contributions. Motivated by these observations, we take an initial step toward this problem by proposing MAPLE. While this work is preliminary in nature, it aims to highlight a promising direction and establish a foundation for future investigation. In particular, this paper makes the following contributions: (i) We propose a scalable framework for constructing tree ensembles that naturally extends from individual decision trees to forests. (ii) We introduce a novel way of leveraging LLMs through multi-agent priors, enabling the integration of diverse perspectives into the ensemble learning process. (iii) We present empirical results on tabular benchmarks that suggest the potential effectiveness of the proposed approach compared to strong tree-based baselines.

2 METHODOLOGY

2.1 PRELIMINARIES

Problem Setup. We consider supervised learning on tabular data. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ denote a dataset with feature vectors $x_i \in \mathbb{R}^d$ and targets $y_i \in \mathcal{Y}$, where $\mathcal{Y}$ is either a finite label set (classification) or $\mathbb{R}$ (regression). Our goal is to construct an ensemble of decision trees $\mathcal{E} = \{T_k\}_{k=1}^K$ that achieves strong generalization performance by combining multiple weak learners while explicitly controlling exploration, feature usage, and diversity.

Bagging and Out-of-Bag Evaluation. Each tree is trained on a bootstrap sample of $\mathcal{D}$. Samples not selected in the bootstrap form the corresponding out-of-bag (OOB) set, which provides an unbiased estimator of generalization error. MAPLE adopts a pure bagging regime, where all available data are used for training and model selection relies exclusively on OOB evaluation.

2.2 ENSEMBLE CONSTRUCTION

MAPLE constructs the ensemble sequentially. At iteration $t \in \{1, \ldots, K\}$, a decision tree T_t is trained via CART on a bootstrap sample using a restricted feature subset $F_t \subseteq \{1, \ldots, d\}$. Unlike standard Random Forests, which sample features uniformly at random, MAPLE draws F_t from an adaptive prior $p_t(\cdot)$ that is initialized by an LLM and updated online based on past performance.

For a tree trained at iteration t, predictive quality is assessed on OOB samples via a task-specific loss ℓ_t. In classification, ℓ_t is log-loss; in regression, ℓ_t is a normalized R^2 so that scores are comparable across OOB subsets. These OOB estimates yield a scalar reward r_t that both drives adaptive exploration and later determines the relative importance of trees in the ensemble.

2.3 MULTI-AGENT PRIOR LEARNING

MAPLE frames feature selection as a multi-agent decision process. A set of A agents is maintained, where each agent a is associated with a feature-selection prior $\pi_a \in \Delta^{d-1}$ encoding a belief over informative features. Each π_a is obtained by prompting a LLM with dataset metadata (e.g., feature names, semantic descriptions, and task specification). At iteration t, a bandit policy selects an agent $a_t \in \{1, \ldots, A\}$, and the corresponding prior is used to sample a feature subset.

Feature sampling is performed using tempering and uniform regularization:

$$p_t(f \mid a_t) = (1 - \lambda) \frac{\left(\pi_{a_t}(f) + \varepsilon\right)^{1/\tau_t}}{\sum_{f'} \left(\pi_{a_t}(f') + \varepsilon\right)^{1/\tau_t}} \; + \; \lambda \frac{1}{d}, \tag{1}$$

where τ_t controls exploration through temperature annealing, $\lambda \in [0, 1]$ prevents premature concentration of probability mass, and $\varepsilon > 0$ is a small smoothing constant ensuring numerical stability. A decision tree T_t is trained using only the sampled feature subset.

The tree is evaluated on OOB samples to obtain a reward $r_t \in [0, 1]$. To emphasize robustness on difficult examples, MAPLE introduces a hardness-aware reward

$$\tilde{r}_t = r_t\big(1 + \bar{h}_t\big), \tag{2}$$

where $\bar{h}_t$ denotes the mean normalized prediction error on OOB samples. The bandit updates its action values using $\tilde{r}_t$, while feature priors are updated by attributing this reward to features actively used by the tree. Let $U_t \subseteq F_t$ denote the set of features with nonzero importance in T_t. The prior update rule for agent a_t is

$$\pi_{a_t}^{(t+1)}(f) = (1 - \eta)\, \pi_{a_t}^{(t)}(f) \;+\; \eta\, \mathbb{I}[f \in U_t]\, \tilde{r}_t, \tag{3}$$

followed by normalization, where η is a learning rate and $\mathbb{I}[\cdot]$ denotes the indicator function.

Proposition 1 (Hierarchical PAC-Bayes bound for learned feature priors) *Let $S = \{z_i\}_{i=1}^N \sim \mathcal{D}^N$ be i.i.d. data and let $\ell(h, z) \in [0, 1]$ be a bounded loss. Fix a hypothesis class $\mathcal{H}$ of decision trees (e.g., bounded depth). Let $\Theta := \{1, \ldots, A\} \times \Delta^{d-1}$ index an agent a and a feature prior π. Assume a data-independent hyperprior H over Θ, for instance*

$$H(a, \pi) := \rho(a)\, H_0(\pi), \qquad \rho(a) = \frac{1}{A}. \tag{4}$$

For each $\theta = (a, \pi) \in \Theta$, let P_θ be a data-independent prior over $\mathcal{H}$ that encodes the inductive bias induced by π (e.g., split-feature propensities). Consider any (data-dependent) joint posterior of the factorized form

$$Q(\theta, h) := R(\theta)\, Q_\theta(h), \tag{5}$$

where R is a hyperposterior over Θ and Q_θ is a posterior over trees. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the draw of S,

$$\mathrm{kl}\Big(\widehat{L}_S(Q) \,\Big\|\, L_{\mathcal{D}}(Q)\Big) \;\le\; \frac{\mathrm{KL}(R \,\|\, H) + \mathbb{E}_{\theta\sim R}[\mathrm{KL}(Q_\theta \,\|\, P_\theta)] + \ln\frac{2\sqrt{N}}{\delta}}{N}, \tag{6}$$

where

$$\widehat{L}_S(Q) := \mathbb{E}_{(\theta,h)\sim Q}\Big[\frac{1}{N}\sum_{i=1}^{N} \ell(h, z_i)\Big], \qquad L_{\mathcal{D}}(Q) := \mathbb{E}_{(\theta,h)\sim Q}\, \mathbb{E}_{z\sim\mathcal{D}}[\ell(h, z)]. \tag{7}$$

Moreover, the hyper-complexity decomposes as

$$\mathrm{KL}(R \,\|\, H) = \mathrm{KL}(R_A \,\|\, \rho) + \mathbb{E}_{a\sim R_A}\big[\mathrm{KL}\big(R_{\pi|a} \,\big\|\, H_0\big)\big]\,, \tag{8}$$

and in particular $\mathrm{KL}(R_A \| \rho) \le \ln A$ when ρ is uniform.

Proof. Apply a standard PAC-Bayes–kl inequality (Maurer, 2004) on the *product space* $\Theta \times \mathcal{H}$ with prior $\widetilde{P}(\theta, h) := H(\theta)P_\theta(h)$ and posterior $\widetilde{Q}(\theta, h) := R(\theta)Q_\theta(h)$. The KL term then decomposes as

$$\mathrm{KL}(\widetilde{Q} \| \widetilde{P}) = \sum_\theta \int R(\theta)Q_\theta(h) \log \frac{R(\theta)Q_\theta(h)}{H(\theta)P_\theta(h)}\, dh \tag{9}$$

$$= \mathrm{KL}(R \| H) + \mathbb{E}_{\theta\sim R}\big[\mathrm{KL}(Q_\theta \| P_\theta)\big], \tag{10}$$

which yields the stated bound. The final decomposition follows from the factorization $H(a, \pi) = \rho(a)H_0(\pi)$ and the chain rule for KL.

Discussion. Proposition 1 shows how multi-agent priors distribute complexity between agent selection (diversity across inductive biases) and within-agent prior adaptation (specialization), aligning with MAPLE's bandit updates. In particular, LLM-derived priors aim to reduce the complexity term by making good posteriors "closer" to their corresponding priors (smaller $\mathrm{KL}(Q_\theta \| P_\theta)$), while MAPLE's online prior updates trade off empirical gains against the explicit hyper-complexity penalty $\mathrm{KL}(R_{\pi|a} \| H_0)$. From this lens, the hyperparameters λ, τ_t, and η can be seen as controls on how aggressively the method adapts priors: they regulate how quickly probability mass concentrates and thus how quickly MAPLE is willing to pay KL complexity for specialization.

Methods	adult	breast	credit	diabetes	heart	liver	phoneme	vehicle
				$K = 50, d = 3$				
Random Forest	.7904 ± .0069	.9678 ± .0014	.7440 ± .0126	.7733 ± .0221	.7872 ± .0184	.7472 ± .0097	.7919 ± .0021	.9564 ± .0061
+ MAPLE (1 agent)	.8058 ± .0021	.9754 ± .0034	.7356 ± .0193	.7965 ± .0158	.7778 ± .0229	.7499 ± .0104	.7952 ± .0024	.9636 ± .0071
+ MAPLE (3 agents)	.8093 ± .0030	.9743 ± .0037	.7441 ± .0180	.7954 ± .0119	.7831 ± .0196	.7464 ± .0089	.7961 ± .0029	.9619 ± .0057
				$K = 100, d = 3$				
Random Forest	.7920 ± .0048	.9678 ± .0014	.7432 ± .0160	.7814 ± .0131	.7803 ± .0168	.7432 ± .0093	.7929 ± .0039	.9593 ± .0040
+ MAPLE (1 agent)	.8105 ± .0008	.9732 ± .0037	.7516 ± .0151	.7936 ± .0133	.7894 ± .0269	.7481 ± .0078	.7965 ± .0024	.9569 ± .0061
+ MAPLE (3 agents)	.8102 ± .0024	.9746 ± .0043	.7560 ± .0146	.7973 ± .0103	.7911 ± .0131	.7522 ± .0167	.7975 ± .0029	.9627 ± .0048

Table 1: **Test balanced accuracy scores (higher is better) for classification tasks.** Values denote mean ± standard deviation over 5 different train/validation splits, each repeated 3 times

Methods	abalone	bike	brazilian	cars	cpu	house	moneyball	naval
				$K = 100, d = 3$				
Random Forest	.4716 ± .0019	.4770 ± .0010	.9313 ± .0009	.8882 ± .0061	.9366 ± .0067	.6920 ± .0028	.8941 ± .0015	.2477 ± .0033
+ MAPLE (1 agent)	.4734 ± .0021	.4910 ± .0005	.9350 ± .0005	.8927 ± .0043	.9448 ± .0041	.7226 ± .0019	.9044 ± .0013	.2648 ± .0025
+ MAPLE (3 agents)	.4730 ± .0022	.4925 ± .0009	.9418 ± .0010	.8948 ± .0046	.9480 ± .0019	.7270 ± .0018	.9079 ± .0015	.2659 ± .0030
				$K = 100, d = 4$				
Random Forest	.5182 ± .0017	.5615 ± .0004	.9593 ± .0015	.9310 ± .0015	.9550 ± .0067	.7427 ± .0022	.9159 ± .0010	.3408 ± .0045
+ MAPLE (1 agent)	.5196 ± .0021	.5681 ± .0004	.9607 ± .0004	.9311 ± .0008	.9591 ± .0034	.7704 ± .0017	.9158 ± .0018	.3552 ± .0017
+ MAPLE (3 agents)	.5232 ± .0019	.5703 ± .0012	.9606 ± .0006	.9354 ± .0014	.9592 ± .0039	.7761 ± .0025	.9202 ± .0014	.3586 ± .0023

Table 2: **Test R^2 scores (higher is better) for regression tasks.** Values denote mean ± standard deviation over 5 different train/validation splits, each repeated 3 times.

2.4 Tree Weighting and Diversity Control

After constructing K trees $\{T_k\}_{k=1}^K$, MAPLE forms the final predictor as a *weighted ensemble*. Let $\widehat{L}_S(T_k)$ denote the empirical OOB loss of tree T_k, and let d_k denote a normalized diversity score measuring its disagreement with the rest of the ensemble. We define a scalar utility

$$s_k \;=\; -(1-\alpha)\,\widehat{L}_S(T_k) \;+\; \alpha\, d_k, \tag{11}$$

where $\alpha \in [0, 1]$ trades off predictive accuracy and diversity.

Tree weighting induces a discrete posterior $\nu \in \Delta^{K-1}$ over the realized hypotheses,

$$\nu_k = (1-\gamma)\,\frac{\exp(s_k/T)}{\sum_{j=1}^K \exp(s_j/T)} \;+\; \gamma\,\frac{1}{K}, \tag{12}$$

with temperature T and uniform mixing coefficient γ. The resulting ensemble predictor is the deterministic mixture $\bar{h}_\nu(x) = \sum_{k=1}^K \nu_k\, T_k(x)$.

Discussion. Specializing Proposition 1 to the finite ensemble $\mathcal{E} = \{T_k\}_{k=1}^K$ with prior ν^{uni} and posterior ν yields the tree-level PAC-Bayes control in Eq. equation 30 (Appendix A.3). Under convex losses, Jensen's inequality gives

$$\widehat{L}_S(\bar{h}_\nu) \;\le\; \widehat{L}_S(\nu), \qquad L_{\mathcal{D}}(\bar{h}_\nu) \;\le\; L_{\mathcal{D}}(\nu). \tag{13}$$

Hence the deterministic ensemble inherits the Gibbs bound through Eq. equation 33. In particular, minimizing the empirical Gibbs risk $\widehat{L}_S(\nu)$ while controlling $\mathrm{KL}(\nu\|\nu^{\text{uni}})$ yields direct control of $L_{\mathcal{D}}(\bar{h}_\nu)$. Consequently, Eq. equation 12 can be interpreted as selecting a posterior over $\mathcal{E}$ that balances empirical utility, complexity regularization, and ensemble diversity.

3 Experiments

Experimental Setup. We evaluate MAPLE on a collection of tabular classification and regression datasets from OpenML (Bischl et al., 2021). Feature priors are generated using `gpt-4o-mini`, a cost-efficient language model that provides sufficiently informative guidance at low expense. MAPLE requires only one LLM call per agent, independent of the ensemble size, making the approach scalable and inexpensive even when multiple agents are used.

Tables 1 and 2 report test balanced accuracy and R^2, respectively. For fair comparison, all methods use the same number of trees K and a fixed maximum depth d for every tree. MAPLE is applied as a drop-in enhancement on top of Random Forest, with either one or multiple agents. Across both classification and regression tasks, MAPLE consistently outperforms the Random Forest baseline. Using multiple agents generally yields larger gains, suggesting that multi-agent prior learning improves feature exploration and ensemble diversity under controlled model capacity.

4 CONCLUSION

We introduced MAPLE, a scalable framework for incorporating LLM-derived feature priors into the construction of tree ensembles via a multi-agent prior learning mechanism. Across tabular benchmarks, MAPLE improves predictive performance and robustness over prior-agnostic baselines while retaining the efficiency and practicality of classical ensemble methods.

Limitations. First, MAPLE relies on the quality and calibration of LLM-generated priors; poorly specified metadata or domain mismatch can yield uninformative or biased feature preferences. Second, our current instantiation considers a fixed depth and a bagging-only regime, leaving open how priors interact with other sources of randomness (e.g., split criteria, subsampling rates) and with alternative ensemble paradigms. Third, the prior update mechanism attributes reward to features used by a tree, which may over-credit correlated features and under-credit features that are useful only through interactions. Finally, while MAPLE is designed to be inexpensive in LLM usage (one call per agent), prompt design and agent diversity can still materially affect results.

Future Work. An immediate direction is to extend MAPLE to boosting-style settings and to study how learned priors should be scheduled as the ensemble focuses on hard examples. Another direction is to develop stronger theoretical guarantees (e.g., regret and generalization bounds) that explicitly account for data-dependent prior updates and multi-agent selection. We also plan to investigate more faithful credit assignment schemes using split-level statistics or Shapley-style attributions, as well as mechanisms for enforcing diversity across agents and across trees. Finally, exploring richer metadata (e.g., feature provenance, units, ontologies) and evaluating MAPLE on larger and more heterogeneous real-world tabular tasks would clarify when LLM-derived priors are most beneficial.

REFERENCES

Nikhil Abhyankar, Parshin Shojaee, and Chandan K Reddy. Llm-fe: Automated feature engineering for tabular data with llms as evolutionary optimizers. *arXiv preprint arXiv:2503.14434*, 2025.

Bernd Bischl, Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael Gomes Mantovani, Jan van Rijn, and Joaquin Vanschoren. Openml benchmarking suites. In J. Vanschoren and S. Yeung (eds.), *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/file/c7e1249ffc03eb9ded908c236bd1996d-Paper-round2.pdf`.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7499–7519, 2024. doi: 10.1109/TNNLS.2022.3229161.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35: 507–520, 2022.

Sungwon Han, Jinsung Yoon, Sercan Ö Arik, and Tomas Pfister. Large language models can automatically engineer features for few-shot tabular learning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 17454–17479, 2024.

Xinrui He, Yikun Ban, Jiaru Zou, Tianxin Wei, Curtiss Cook, and Jingrui He. Llm-forest: Ensemble learning of llms with graph-augmented prompts for data imputation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6921–6936, 2025.

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 *International conference on artificial intelligence and statistics*, pp. 5549–5581. PMLR, 2023.

Noah Hollmann, Samuel Müller, and Frank Hutter. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. *Advances in Neural Information Processing Systems*, 36:44753–44775, 2023.

Daniel P Jeong, Zachary Chase Lipton, and Pradeep Kumar Ravikumar. Llm-select: Feature selection with large language models. *Transactions on Machine Learning Research*, 2024.

Ricardo Knauer, Mario Koddenbrock, Raphael Wallsberger, Nicholas M Brisson, Georg N Duda, Deborah Falla, David W Evans, and Erik Rodner. 'oh llm, i'm asking thee, please give me a decision tree': Zero-shot decision tree induction and embedding with large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1196–1206, 2025.

Tennison Liu, Nicolas Huynh, and Mihaela van der Schaar. Decision tree induction through llms via semantically-aware evolution. In *The Thirteenth International Conference on Learning Representations*, 2025.

Andreas Maurer. A note on the pac bayesian theorem. *arXiv preprint cs/0411099*, 2004.

Jaehyun Nam, Kyuyoung Kim, Seunghyuk Oh, Jihoon Tack, Jaehyung Kim, and Jinwoo Shin. Optimized feature generation for tabular data via llms with decision tree reasoning. *Advances in Neural Information Processing Systems*, 37:92352–92380, 2024.

Nabeel Seedat, Nicolas Huynh, Boris Van Breugel, and Mihaela Van Der Schaar. Curated llm: Synergy of llms and data curation for tabular augmentation in low-data regimes. In *International Conference on Machine Learning*, pp. 44060–44092. PMLR, 2024.

Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2021.11.011. URL https://www.sciencedirect.com/science/article/pii/S1566253521002360.

Hangting Ye, Jinmeng Li, He Zhao, Dandan Guo, and Yi Chang. Llm meeting decision trees on tabular data. *arXiv preprint arXiv:2505.17918*, 2025.

Xinhao Zhang, Jinghan Zhang, Banafsheh Rekabdar, Yuanchun Zhou, Pengfei Wang, and Kunpeng Liu. Dynamic and adaptive feature generation with llm. *arXiv preprint arXiv:2406.03505*, 2024.

A Appendix

A.1 System Prompt

System prompts define the role and reasoning style assigned to each agent before it receives dataset-specific content. They shape how each agent interprets feature descriptions and sets prior weights, encouraging complementary inductive biases. We use up to five agent-specific system prompts (balanced, domain-focused, statistical, conservative, exploratory), each with a fixed temperature.

From an optimization viewpoint, this design also aligns with the hierarchical analysis in Proposition 1: different system prompts instantiate different initial inductive biases, and subsequent data-driven adaptation refines each bias under an explicit complexity budget. Intuitively, the prompts control *where* MAPLE starts in prior space, whereas bandit selection and posterior reweighting control *how far* it moves based on observed OOB rewards.

Balanced Agent System Prompt

You are an AutoML prior designer for decision-tree models. Given dataset context and feature descriptions, propose a weight in $[0, 1]$ for each feature. Higher weight means stronger prior to prefer splits using that feature. Use the full scale $[0, 1]$ when appropriate. Be balanced and consider all aspects.

Domain-Focused Agent System Prompt

You are a domain expert designing feature importance priors for decision trees. Focus on domain knowledge and causal relationships. Weight features based on their likely causal or predictive relationship with the target. Strongly prefer features with clear domain justification.

Statistical Agent System Prompt

You are a statistician designing feature priors for decision trees. Focus on statistical properties: variance, potential for separation, information content. Prefer features that likely have high information gain or good split potential.

Conservative Agent System Prompt

You are a conservative AutoML designer. Be cautious with feature weights - only give high weights to features with very strong evidence. Prefer simpler, more interpretable features. Penalize high-cardinality or noisy features.

Exploratory Agent System Prompt

You are an exploratory AutoML designer willing to try unconventional feature combinations. Don't be afraid to give high weights to features that might seem less obvious. Look for hidden patterns and non-obvious predictors.

A.2 User Prompt

The user prompt provides the task-specific context that every agent uses to produce feature weights. It includes dataset identity, task and target information, and feature-level descriptions, together with an explicit scoring rubric and output rules. Each agent receives the same user prompt and returns a weight in $[0, 1]$ for every feature. Missing features trigger up to n_{trials} retries; otherwise a uniform prior (0.5) is used as fallback.

User Prompt Template for Prior Elicitation

DATASET: {`name`}
TASK: {`task`} with target {`label`}
Feature descriptions:
– Feature: {`f_1`} — Description: {`desc_1`}
– Feature: {`f_2`} — Description: {`desc_2`}
...
Your task: Assign a weight w in $[0, 1]$ to *each* feature.
Weighting rubric:

- 0.90–1.00: direct proxy/causal driver of the target
- 0.70–0.80: strong predictor with solid domain rationale
- 0.50–0.60: moderately informative
- 0.30–0.40: weak or ambiguous relation
- 0.10–0.20: marginal relevance

Guidelines:

- Provide exactly one weight for every listed feature
- Weights need not sum to 1

A.3 Detailed Theoretical Analysis

This subsection provides a notation-consistent derivation of the tree-weight posterior and clarifies how Jensen's inequality transfers the PAC-Bayes control from the Gibbs predictor to the deterministic ensemble predictor. Following the main text, the realized ensemble is denoted by $\mathcal{E} = \{T_k\}_{k=1}^{K}$.

Entropy-regularized variational objective on $\mathcal{E}$. The posterior $\nu \in \Delta^{K-1}$ in Eq. equation 12 is a distribution over trees in $\mathcal{E}$, i.e., over indices $k \in \{1, \ldots, K\}$. Define the uniform reference measure on $\mathcal{E}$ by

$$\nu_k^{\text{uni}} = \frac{1}{K}, \qquad k = 1, \ldots, K. \tag{14}$$

Given utilities from Eq. equation 11,

$$s_k = -(1-\alpha)\, \widehat{L}_S(T_k) + \alpha\, d_k, \tag{15}$$

consider the objective

$$\nu^\star = \arg\max_{\nu \in \Delta^{K-1}} \left\{ \sum_{k=1}^{K} \nu_k s_k - T\, \mathrm{KL}(\nu \| \nu^{\text{uni}}) \right\}, \tag{16}$$

with $T > 0$. The expected-utility term favors accurate and diverse trees (through s_k), while the KL term controls concentration and prevents unstable over-commitment to a small subset of trees when OOB estimates are noisy. The objective is strictly concave on the simplex, so the optimizer is unique.

Using

$$\mathcal{L}(\nu, \lambda_0) = \sum_{k=1}^{K} \nu_k s_k - T \sum_{k=1}^{K} \nu_k \log \frac{\nu_k}{\nu_k^{\text{uni}}} + \lambda_0 \left(\sum_{k=1}^{K} \nu_k - 1 \right), \tag{17}$$

stationarity gives

$$0 = s_k - T \left(\log \frac{\nu_k}{\nu_k^{\text{uni}}} + 1 \right) + \lambda_0, \tag{18}$$

and therefore

$$\nu_k \propto \nu_k^{\text{uni}} \exp\left(\frac{s_k}{T} \right). \tag{19}$$

Substituting $\nu_k^{\mathrm{uni}} = 1/K$ and normalizing yields

$$\nu_k^{\star} = \frac{\exp(s_k/T)}{\sum_{j=1}^{K} \exp(s_j/T)}. \tag{20}$$

Hence the softmax/Gibbs weighting is the variational solution of Eq. equation 16 under the ensemble notation used in the main text.

Uniform smoothing and complexity attenuation. MAPLE uses

$$\nu = (1-\gamma)\nu^{\star} + \gamma\nu^{\mathrm{uni}}, \qquad \gamma \in [0,1], \tag{21}$$

which is equivalent to Eq. equation 12. This interpolation keeps every tree active ($\nu_k \geq \gamma/K$), improves robustness to OOB noise, and contracts complexity relative to ν^{uni}. By convexity of KL in its first argument,

$$\mathrm{KL}(\nu\|\nu^{\mathrm{uni}}) = \mathrm{KL}\big((1-\gamma)\nu^{\star} + \gamma\nu^{\mathrm{uni}} \,\big\|\, \nu^{\mathrm{uni}}\big) \tag{22}$$

$$\leq (1-\gamma)\,\mathrm{KL}(\nu^{\star}\|\nu^{\mathrm{uni}}) + \gamma\,\mathrm{KL}(\nu^{\mathrm{uni}}\|\nu^{\mathrm{uni}}) \tag{23}$$

$$= (1-\gamma)\,\mathrm{KL}(\nu^{\star}\|\nu^{\mathrm{uni}}). \tag{24}$$

Thus, γ is an explicit exploration–concentration knob and also a direct complexity-control parameter from a PAC-Bayes perspective.

Jensen step and PAC-Bayes upper-bound chain. Let $\widetilde{T} \sim \nu$ be a random tree sampled from $\mathcal{E}$ (Gibbs predictor), and define the deterministic weighted predictor

$$\bar{h}_{\nu}(x) = \mathbb{E}_{\widetilde{T}\sim\nu}[\widetilde{T}(x)] = \sum_{k=1}^{K} \nu_k T_k(x). \tag{25}$$

Assume $\ell(\cdot, y)$ is convex in its prediction argument for each y. Then for every (x, y),

$$\ell(\bar{h}_{\nu}(x), y) = \ell\Big(\mathbb{E}_{\widetilde{T}\sim\nu}[\widetilde{T}(x)], y\Big) \tag{26}$$

$$\leq \mathbb{E}_{\widetilde{T}\sim\nu}[\ell(\widetilde{T}(x), y)] = \sum_{k=1}^{K} \nu_k \ell(T_k(x), y). \tag{27}$$

Averaging over S and over $\mathcal{D}$ gives

$$\widehat{L}_S(\bar{h}_{\nu}) \leq \widehat{L}_S(\nu), \qquad L_{\mathcal{D}}(\bar{h}_{\nu}) \leq L_{\mathcal{D}}(\nu), \tag{28}$$

where

$$\widehat{L}_S(\nu) := \mathbb{E}_{\widetilde{T}\sim\nu}\Big[\frac{1}{N}\sum_{i=1}^{N} \ell(\widetilde{T}, z_i)\Big], \qquad L_{\mathcal{D}}(\nu) := \mathbb{E}_{\widetilde{T}\sim\nu}\,\mathbb{E}_{z\sim\mathcal{D}}[\ell(\widetilde{T}, z)]. \tag{29}$$

Applying the same PAC-Bayes–kl argument as in Proposition 1, now on the finite class induced by $\mathcal{E}$ with prior ν^{uni} and posterior ν, yields (with probability at least $1-\delta$)

$$\mathrm{kl}\Big(\widehat{L}_S(\nu) \,\Big\|\, L_{\mathcal{D}}(\nu)\Big) \leq \frac{\mathrm{KL}(\nu\|\nu^{\mathrm{uni}}) + \ln\frac{2\sqrt{N}}{\delta}}{N} =: \varepsilon_{\nu}(\delta). \tag{30}$$

Equivalently,

$$L_{\mathcal{D}}(\nu) \leq \mathrm{kl}_{+}^{-1}\big(\widehat{L}_S(\nu), \varepsilon_{\nu}(\delta)\big), \tag{31}$$

where

$$\mathrm{kl}_{+}^{-1}(q, \epsilon) := \sup\{p \in [q, 1] : \mathrm{kl}(q\|p) \leq \epsilon\}. \tag{32}$$

Combining with Eq. equation 28 gives the explicit chain

$$L_{\mathcal{D}}(\bar{h}_{\nu}) \leq L_{\mathcal{D}}(\nu) \leq \mathrm{kl}_{+}^{-1}\big(\widehat{L}_S(\nu), \varepsilon_{\nu}(\delta)\big), \tag{33}$$

which is the desired upper bound for the deterministic weighted ensemble predictor constructed from $\mathcal{E}$.