Automatic Machine Learning by Pipeline Synthesis using Model-Based Reinforcement Learning and a Grammar

Iddo Drori, Yamuna Krishnamurthy, Raoni de Paula Lourenco, Remi Rampin, Roque Lopez, Kyunghyun Cho, Claudio Silva, Juliana Freire

NEW YORK UNIVERSITY

VIDA VISUALIZATION IMAGING AND DATA ANALYSIS CENTER

AlphaD3M Goals

Strongest AutoML systems are based on neural networks, evolutionary algorithms, and Bayesian optimization. Recently, AlphaD3M reached SOA results with order of magnitude speedup using reinforcement learning with self-play. We extend AlphaD3M using a pipeline grammar and generalize from many datasets and similar tasks by a pre-trained model. Results demonstrate improved performance compared with existing methods on AutoML benchmark datasets.

AlphaD3M Pipeline Grammar

Table 1: Grammar $< T, N, P, S >$ for machine learning pipelines for a classification task.

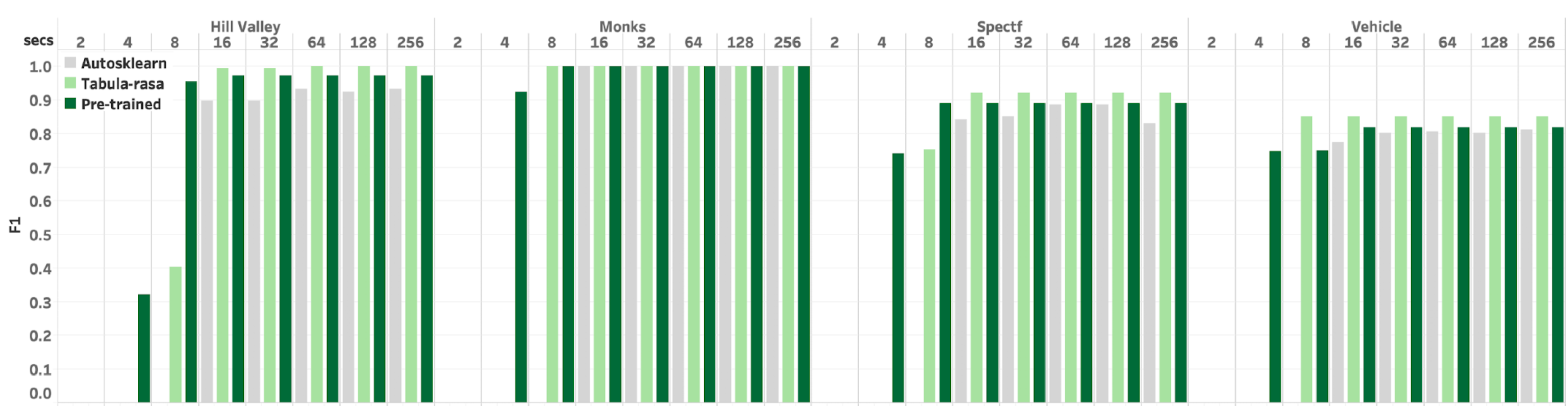
T[Terminals]	$SkImputer$, $MissingIndicator$, $OneHotEncoder$, $OrdinalEncoder$, PCA ..., $GaussianNB$, $RidgeClassifier$, $SGDClassifier$, $LinearSVC$
N[Non-Terminals]	$DataCleaning$ <DC>, $DataTransformation$ <DT>, $Estimators$ <E>
S[Start]	S
P[Production Rules]	<S> ::= <E> | <DC> <E> | <DT> <E> | <DC> <DT> <E> <DC> ::= $SkImputer$ <DC> | ... | $MissingIndicator$ <DC> | $SkImputer$ | ... | $MissingIndicator$ <DT> ::= $OneHotEncoder$ <DT> $OrdinalEncoder$ <DT> | ... | PCA <DT> | $OneHotEncoder$ $OrdinalEncoder$ | ... | PCA <E> ::= $GaussianNB$ | $RidgeClassifier$ | $SGDClassifier$ | ... | $LinearSVC$

Algorithm 1 Pipeline State Encoding

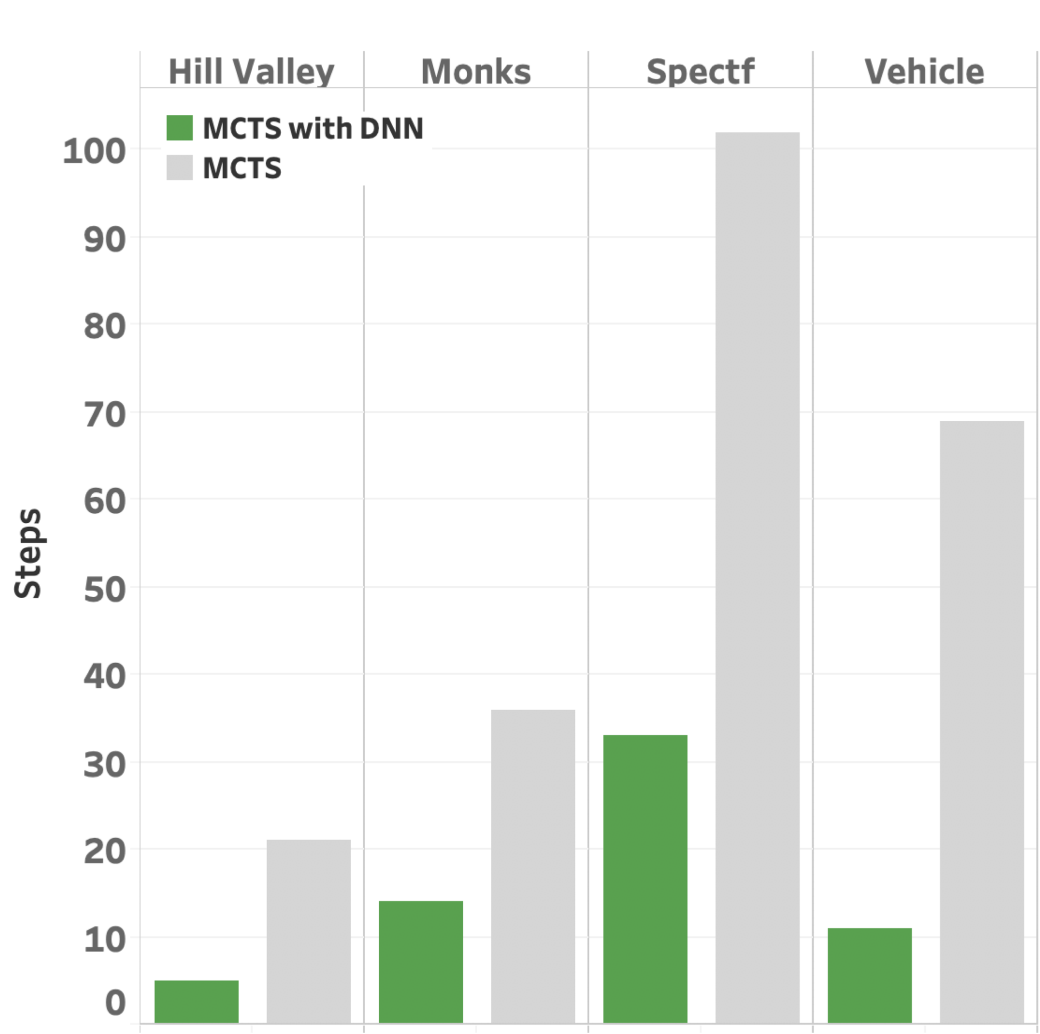
Given datasets D, tasks T, and a set of possible pipeline sequences $S_1, \ldots, S_n$, from the available machine learning, and data pre and post processing primitives.

- For each dataset D_i and task T_j:
  1. Encode dataset D_i as meta data features $f(D_i)$.
  2. Encode task T_j.
  3. Encode the current pipeline at time t by a vector S_t.
  4. Encode action $f_a(S_t)$, so policy π maps $(f(D_i), T_j, S_t)$ to $f_a(S_1), \ldots, f_a(S_n)$.

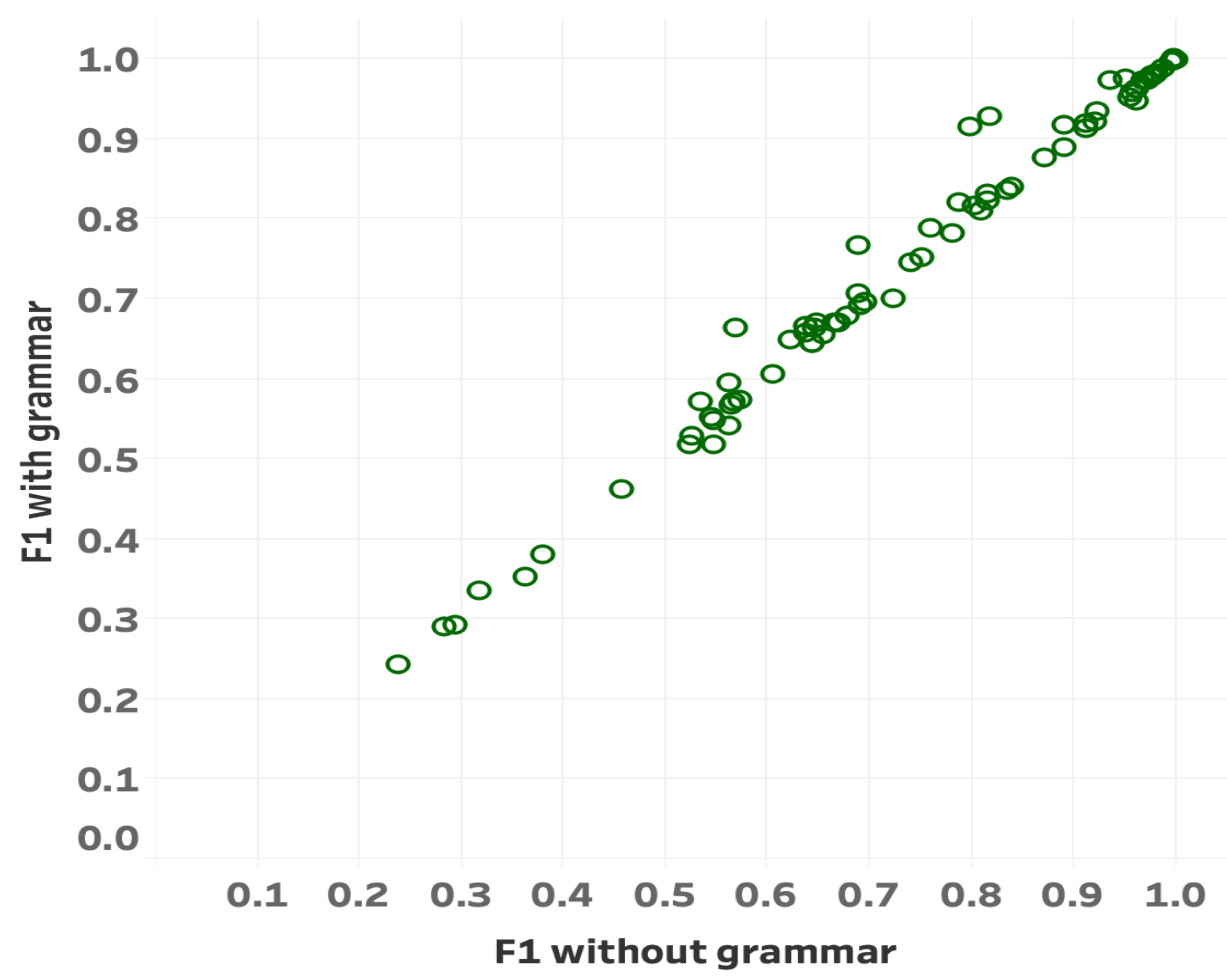
AlphaD3M Performance Comparison using Sklearn Primitives



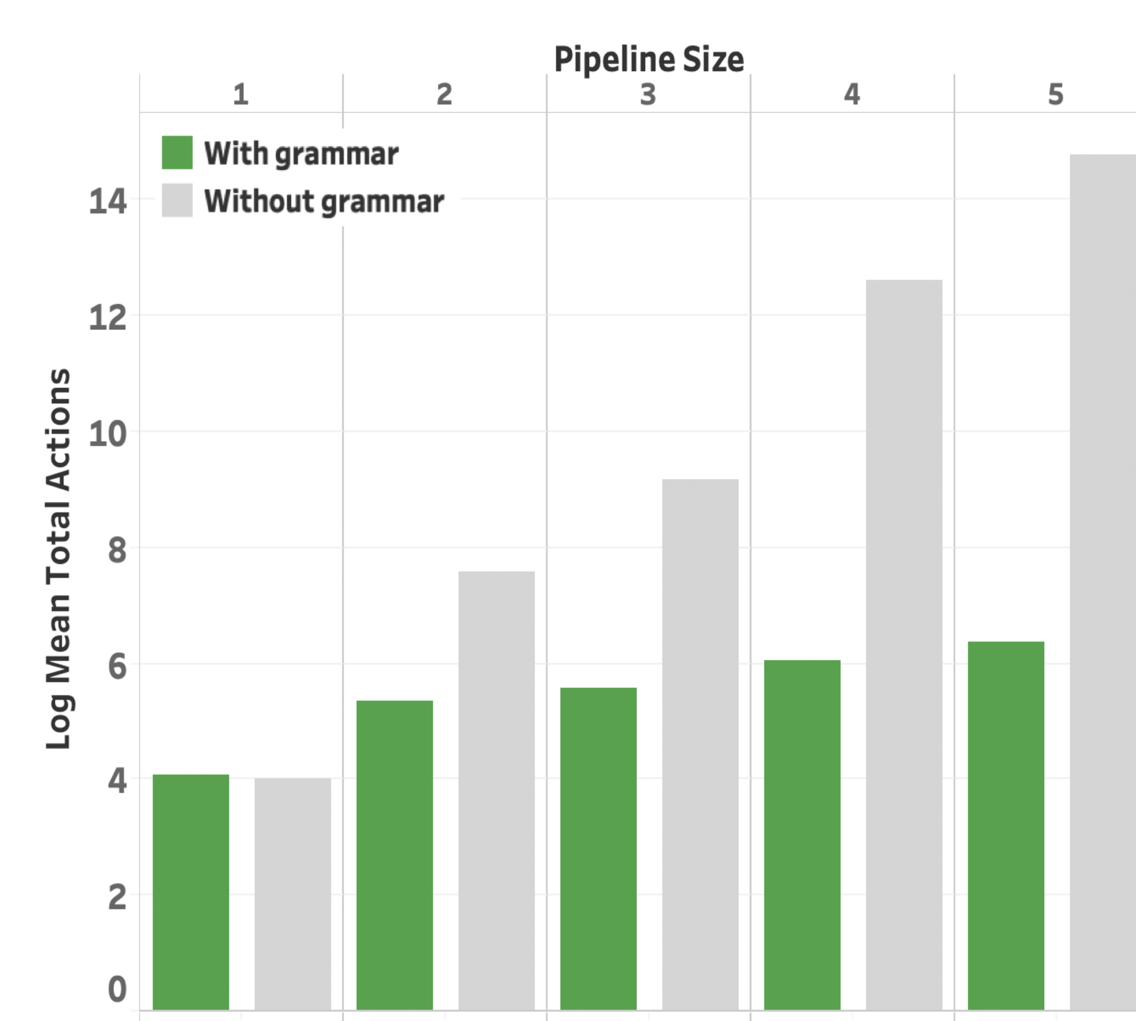
Performance comparison between AlphaD3M using a grammar pre-trained on other datasets (dark green), AlphaD3M using a grammar trained from scratch (light green), and AutoSklearn (gray). Vertical axis is f1-score, time in seconds is horizontal axis.



Comparison of the number of evaluation steps of MCTS with a neural network (green) vs. MCTS only (gray).



Comparison of performance using a pipeline grammar vs. without using a pipeline grammar: each point represents a different OpenML dataset. Performance is not degraded even though computation time is reduced.



Comparison of log mean total actions with and without a pipeline grammar

References

Drori, I., Krishnamurthy, Y., Rampin, R., Lourenco, R., Ono, J.P., Cho, K., Silva, C., and Freire, J. **AlphaD3M: Machine Learning Pipeline Synthesis.** In AutoML Workshop at ICML 2018.

Drori, I., Krishnamurthy, Y., Lourenco, R., Rampin, R., Cho, K., Silva, C., and Freire, J. **Automatic Machine Learning by Pipeline Synthesis using Model-Based Reinforcement Learning.** In AutoML Workshop at ICML 2019.