Appendix

1 DIFFERENCES BETWEEN LATTEBENCH AND ORIGINAL METHODS

In LATTEBench, we focus on comparing and discussing the 6 dimensions of LATTE methods, with particular emphasis on the 4 main dimensions. Therefore, for fair comparison, we standardized the settings of other modules in the pipeline to accurately attribute the causes of changes in performance and overhead. Here, we provide a detailed description in Table 1 of the differences between original methods and the LATTEBench pipeline in terms of these default settings and additional modules.

Table 1. Extended Version of LATTEBench Configuration Space.

Alias	Original Methods	Detailed Differences of Each Method
CGN	SMARTFEAT[9], FeatLLM[5], GPT-Signal [12], FREEFORM [8], RAFG [13]	SMARTFEAT is a user-interactive dialogue system that uses 3 feature selection metrics provided by sklearn. GPT-Signal has no open-source implementation and is also a semi-automatic method involving human participation. FREEFORM and FeatLLM are model ensemble methods directly oriented toward prediction tasks, where the former targets linear classifiers and the latter is designed for genotype data; RAFG leverages RAG for assistance.
CGC		
CGR		
CGN$_h$	FEBias [7], CAAFE [6]	FEBias has no open-source implementation and no selector. CAAFE relies solely on LLMs for feature selection and lacks a selector.
CGC$_h$		
CGR$_h$		
CGN$_t$	*– (New Variant) –*	
CGC$_t$		
CGR$_t$		
TMN$_h$	LFG [14], Adda [10]	LFG lacks a selector and invokes the LLM Agent through multi-turn conversations, which leads to context length overflow issues when dealing with a large number of features or rich metadata. Adda requires pre-training a metadata embedding model on datasets in advance, and leverages UDFs to integrate the LATTE algorithm into the DBMS for acceleration.
TMC$_h$		
TMR$_h$		
TMN	*– (New Variant) –*	
TMC		
TMR		
GGN	LPFG [4], Rouge One [2]	Rouge One does not have an open-source implementation and introduces external knowledge through RAG, thus it is represented by LPFG.
GGC		
GGR		
OGC$_c$	OCTree [11]	No modification to OCTree. FEBP does not have an open-source implementation and lacks detailed descriptions for its implementation. To ensure a fair comparison, we implemented it by modifying the OCTree framework.
OGC	*– (Base) –*	
OGR	FEBP [15]	
EBR$_w$	ELLM-FT [3]	ELLM-FT, as a continuation of the RL method GRFG, only receives numerical tables as input without incorporating metadata and instances, and it also does not do feature selection. LLM-FE does not actually perform evolutionary algorithms but always selects top-k demonstrations; LATTEBench uses the population evolution framework of ELLM-FT as a replacement.
EBR	*– (Base) –*	
EBC	LLM-FE [1]	

2 EXTENDED EXPERIMENT

To further investigate the cost-effectiveness of different LATTE configurations, we tested CoT methods under higher token budget scales (140k+ tokens) and compared their cost-effectiveness with OGC, the best-performing high-cost method under fixed-round settings. The results are shown in Figure 1. We observe that simple CoT methods, even those without demonstrations, consistently outperform OGC.

Observation: *OPRO iteratively optimizes the quality of a single output, which is less cost-effective than multiple independent LLM queries.*

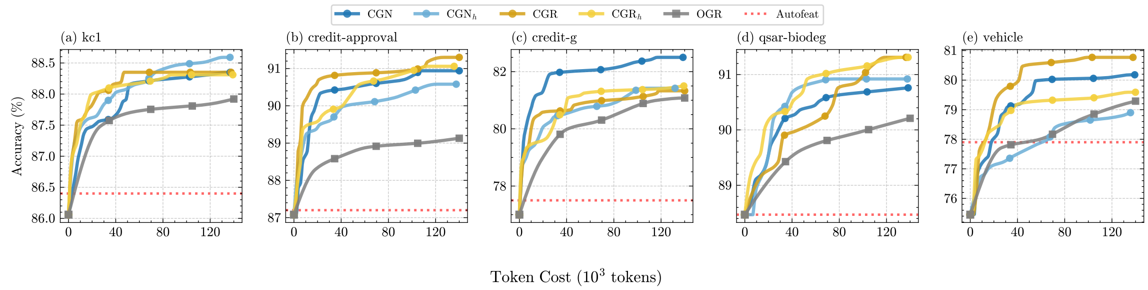

Fig. 1. VG vs. Token Cost for simple CoT methods and OGR with high budgets.

REFERENCES

[1] Nikhil Abhyankar, Parshin Shojaee, and Chandan K Reddy. 2025. LLM-FE: Automated Feature Engineering for Tabular Data with LLMs as Evolutionary Optimizers. *arXiv preprint arXiv:2503.14434* (2025).

[2] Henrik Bradland, Morten Goodwin, Vladimir I Zadorozhny, and Per-Arne Andersen. 2025. Knowledge-Informed Automatic Feature Extraction via Collaborative Large Language Model Agents. *arXiv preprint arXiv:2511.15074* (2025).

[3] Nanxu Gong, Chandan K Reddy, Wangyang Ying, Haifeng Chen, and Yanjie Fu. 2025. Evolutionary large language model for automated feature transformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 16844–16852.

[4] Nanxu Gong, Xinyuan Wang, Wangyang Ying, Haoyue Bai, Sixun Dong, Haifeng Chen, and Yanjie Fu. 2025. Unsupervised Feature Transformation via In-context Generation, Generator-critic LLM Agents, and Duet-play Teaming. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*. 2820–2828.

[5] Sungwon Han, Jinsung Yoon, Sercan Ö. Arik, and Tomas Pfister. 2024. Large language models can automatically engineer features for few-shot tabular learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*. Article 695, 26 pages.

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

[7] Jaris Küken, Lennart Purucker, and Frank Hutter. 2024. Large language models engineer too many simple features for tabular data. *arXiv preprint arXiv:2410.17787* (2024).

[8] Joseph Lee, Shu Yang, Jae Young Baik, Xiaoxi Liu, Zhen Tan, Dawei Li, Zixuan Wen, Bojian Hou, Duy Duong-Tran, Tianlong Chen, et al. 2025. Knowledge-driven feature selection and engineering for genotype data with large language models. *AMIA Summits on Translational Science Proceedings* 2025 (2025), 250.

[9] Yin Lin, Bolin Ding, HV Jagadish, and Jingren Zhou. 2024. SmartFeat: efficient feature construction through feature-level foundation model interactions. *14th Annual Conference on Innovative Data Systems Research* (2024).

[10] Kuan Lu, Zhihui Yang, Sai Wu, Ruichen Xia, Dongxiang Zhang, and Gang Chen. 2025. Adda: Towards Efficient in-Database Feature Generation via LLM-based Agents. *Proceedings of the ACM on Management of Data* 3, 3 (2025), 1–27.

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

[12] Yining Wang, Jinman Zhao, and Yuri Lawryshyn. 2024. GPT-Signal: Generative AI for Semi-automated Feature Engineering in the Alpha Research Process. In *Proceedings of the Eighth Financial Technology and Natural Language Processing and the 1st Agent AI for Scenario Planning*. 42–53.

[13] XinHao Zhang, Jinghan Zhang, Fengran Mo, Yuzhong Chen, and Kunpeng Liu. 2024. Retrieval-Augmented Feature Generation for Domain-Specific Classification. *arXiv preprint arXiv:2406.11177* (2024).

[14] Xinhao Zhang, Jinghan Zhang, Banafsheh Rekabdar, Yuanchun Zhou, Pengfei Wang, and Kunpeng Liu. 2025. Dynamic and Adaptive Feature Generation with LLM. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*. 7029–7037.

[15] Yufeng Zou, Jean Utke, Diego Klabjan, and Han Liu. 2025. Automated Feature Engineering by Prompting. https://openreview.net/forum?id=ZXO7iURZfW