Discrete-LLM-AMC: Token- and Parameter-Efficient In-Context Automatic Modulation Classification with Discretized Signal Statistics

Mohammad Rostami, Atik Faysal, Reihaneh Gh. Roshan, Huaxia Wang, Nikhil Muralidhar, and Yu-Dong Yao

***Abstract*—Large Language Models (LLMs) can perform Automatic Modulation Classification (AMC) in a training-free, open-set manner when equipped with carefully designed in-context prompts [1]. Building on this prior work, we target the practical bottlenecks of long prompt contexts and large model sizes that impede in-the-loop deployment. We present Discrete-LLM-AMC, a token- and parameter-efficient variant that: (i) discretizes higher-order statistics and cumulants into compact symbolic tokens, (ii) prunes the exemplar list via a lightweight top-k neural prefilter and filters misleading/low-impact features using rationales extracted from prior LLM responses, and (iii) enforces label-only predictions through a calibrated prompt template. Together, these changes reduce both input/output tokens and the model parameter footprint by more than half while maintaining competitive accuracy. On synthetic AMC with ten modulation types under noise, a 7B DeepSeek-R1-Distill-Qwen baseline achieves 5.2% accuracy, whereas our system—using an approximately 5B-parameter Gemini 2.5 Flash model—attains 39.0% accuracy. These results demonstrate that careful discretization and context selection can cut inference cost by over 2× while preserving the open-set, training-free advantages of prompt-based AMC and enabling practical in-the-loop use.**

***Index Terms*—Automatic modulation classification, large language models, prompt engineering, higher-order statistics.**

I. Introduction

Need to change results based on the new table(s)

Automatic Modulation Classification (AMC) is a pivotal technology in modern wireless communication systems, underpinning critical applications such as cognitive radio, spectrum sensing, and interference management. The accurate identification of modulation schemes is essential for efficient spectrum utilization and enhancing the adaptability and reliability of communication networks. However, AMC remains a challenging problem due to the complex interplay of signals with ambient noise, interference, and various channel impairments [2]–[4].

Historically, AMC approaches evolved from traditional feature-based methods, which relied on handcrafted signal features like higher-order statistics, to sophisticated deep learning models. Convolutional Neural Networks (CNNs) and, more recently, Transformer-based architectures, have demonstrated strong performance in capturing complex signal dependencies and achieving high classification accuracy, including in low SNR environments [5]–[7]. Self-supervised denoising autoencoders further improve robustness and data efficiency under noise [8]–[10]. Despite these advancements, most deep learning solutions demand extensive labeled datasets for training and often require retraining or fine-tuning for new operating conditions, limiting robustness across diverse noise scenarios and generalization beyond specific tasks.

Recent work advocates a Wireless Physical-layer Foundation Model (WPFM) to replace siloed task models with a general, adaptable backbone [4]. As one practical instantiation, LLM prompting expresses higher-order statistics as text to enable training-free, open-set AMC via one-shot reasoning [1]. However, current LLM-based AMC is costly due to long numeric prompts and large models, limiting in-the-loop edge use.

This letter introduces Discrete – LLM – AMC, a token- and parameter-efficient variant that preserves the advantages of training-free, open-set prompting while cutting inference cost. We discretize higher-order statistics into compact tokens, prune exemplars via a lightweight top-k shortlist, and enforce label-only responses—together reducing input/output tokens and effective parameter footprint by over 2×. On ten-class synthetic AMC under noise, an approximately 5B Gemini 2.5 Flash attains 39.0% accuracy versus 5.2% for a 7B DeepSeek-R1-Distill-Qwen baseline, while remaining competitive with 32B models in noiseless settings at a fraction of the compute.

II. Method

We adopt a three-stage, plug-and-play pipeline adapted from prior framework [1], redesigning each stage for improved efficiency. The pipeline involves: (1) discretizing Inphase/Quadrature (I/Q) signals into compact statistical tokens; (2) assembling a concise prompt using a pruned set of exemplars; and (3) reframing the query to enable constrained decoding.

A. Stage 1: Discrete Statistical Tokens (vs. numeric features)

Given a complex baseband segment, we compute a compact set of descriptive statistics and cumulant-derived features (e.g., skewness, kurtosis, moments, k-stats) as in [1]. Unlike plug-and-play [1], which serialized floating-point values verbatim,

M. Rostami, A. Faysal, H. Wang are with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ, USA (e-mail: {rostami23, faysal24, wanghu}@rowan.edu).

A. Faysal is with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ, USA (e-mail: faysal24@rowan.edu).

R. Gh. Roshan and N. Muralidhar are with the Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ, USA (e-mail: {rghasemi, nmurali1}@stevens.edu).

we map each scalar to one of B bins and emit a short symbolic token per feature. We also remove low-impact fields (e.g., `nobs/min/max/mean/variance`) and include SNR, reducing per-item fields and replacing long decimal strings with short codes. This discretization cuts prompt tokens and standardizes feature scales for robust prompting.

B. Stage 2: Compact Prompt with k-top Exemplar Pruning

We first construct a balanced exemplar pool spanning all classes and SNR levels. For each query, we then select at most K exemplars (default $K \leq 10$) using a lightweight top-k shortlisting model to identify classes most relevant to the query. In contrast to the large, fixed exemplar sets from previous work [1], this dynamic pruning strategy maintains discriminative coverage while substantially reducing the context length. The final prompt is composed of: (i) a concise instruction constraining the model's output to a predefined label set; (ii) the K pruned exemplars represented with discrete tokens; and (iii) the query signal, also in its discretized format.

Shortlisting Classifier: To prune the candidate label space for each query, we train a lightweight visual classifier on signal constellation diagrams. The classifier's architecture is a ViT encoder, initialized with weights from a pretrained autoencoder to facilitate efficient feature extraction. At inference, this model identifies the top-k most probable labels for a given query. These labels are then used to dynamically construct the restricted multiple-choice set in the prompt, acting as a computationally efficient pre-filter that reduces prompt length with negligible impact on class coverage.

C. Stage 3: Improving Prompt Formulation with Constrained Decoding

Finally, to enhance reliability and enable constrained decoding, we employ a structured prompt formulation. The instruction block is refined with more detailed explanations and optimized templates. Crucially, the query block is reframed from an open-ended question into a multiple-choice format, providing the model with an enumerated list of valid class options. While this detailed formulation increases the raw prompt length, its structure allows for more efficient inference. In conjunction with the token-saving measures from Stages 1–2, this methodology yields a net efficiency gain of greater than 2× in token/parameter usage while maintaining competitive accuracy.

A new picture of the pipeline

A new figure of the prompt

III. Experimental Setup

How should we refer to the WOCC paper?

a) Dataset: We adopt the synthetic dataset and evaluation protocol from Rostami et al. [1]. The dataset consists of I/Q signals representing 10 digital modulation types: 4ASK, 4PAM, 8ASK, 16PAM, CPFSK, DQPSK, GFSK, GMSK, OOK, and OQPSK. Each signal is generated across a Signal-to-Noise Ratio (SNR) range of -10 dB to $+10$ dB. All evaluations are performed in a one-shot, in-context learning setting where the model must classify a query signal given a single example of selected class by the shortlisting classifier.

b) Baselines: Our primary baseline is the plug-and-play framework [1], which prompts models with raw floating-point statistical features and a comprehensive, unpruned set of exemplars. To assess its performance, we apply this method to several open-weight models, including `DeepSeek-R1-Distill-Qwen-7B`, and `DeepSeek-R1-Distill-Qwen-32B`. Additionally, we report results from a larger, proprietary model (`o3-mini`) to establish a practical upper bound on performance. We also included results from other transformers-based models, includingthe Nmformer [7] and DenoMAE [8] for comparison.

c) Proposed Method and Models: We evaluate our three-stage pipeline, which integrates discretized statistical tokens, dynamic top-k exemplar pruning via a shortlisting classifier, and a structured multiple-choice prompt format. For our experiments, we use Google's Gemini models, accessed via their public API:

- `Gemini 2.5 Flash`: A highly efficient model optimized for speed and low-cost inference.
- `Gemini 2.5 Pro`: A state-of-the-art, high-performance model.

These models were selected to analyze our method's effectiveness across different points on the performance-efficiency spectrum. Our primary metrics are classification accuracy across the SNR range and the final prompt length in tokens.

IV. Results

Our experimental results validate the effectiveness of the proposed Discrete-LLM-AMC framework, demonstrating a favorable trade-off between model efficiency and classification accuracy across a series of targeted evaluations.

The primary findings, summarized in Table I, highlight the significant advantages of our discretized prompting strategy. The baseline plug-and-play approach, which uses raw numeric features, proves ineffective for smaller models; the 7B DeepSeek model achieves only 5.20% accuracy. In stark contrast, our method enables the even smaller 5B Gemini 2.5 Flash model to reach a competitive 45.41% accuracy. This result is particularly noteworthy as it is comparable to the performance of the much larger 32B DeepSeek model (47.80%) but is achieved with a prompt size of just 1.2K tokens—less than half the 3K+ tokens required by the baseline. Furthermore, when applying our method with the more powerful Gemini 2.5 Pro, accuracy climbs to 69.78%, closely approaching the 69.92% performance of the proprietary 200B-parameter o3-mini model, again with a significantly smaller token and parameter footprint. While specialized supervised models like DenoMAE still hold an edge in absolute accuracy (81.30%), our approach offers the crucial advantages of being entirely training-free and open-set.

We then investigate the critical impact of the exemplar selection strategy on model performance, with results detailed in Table II. A deterministic but naive strategy of selecting exemplars closest to class centroids proved suboptimal, yielding

TABLE I
ACCURACY AND EFFICIENCY SUMMARY (REPRESENTATIVE)

Model	Parameters	# Tokens	Accuracy (%)
Nmformer [7]	-	-	71.60%
DenoMAE [8]	-	-	**81.30%**
DenoMAE2.0 [9]	-	-	82.40%
DeepSeek-R1-Distill-Qwen [1]	7B	3K+	5.20%
DeepSeek-R1-Distill-Qwen [1]	32B	3K+	47.80%
OpenAI's o3-mini [1]	200B	3K+	69.92%
Google's gemini-2.5-flash	5B	1.2K	**45.41%**
Google's gemini-2.5-pro	-	1.4K	**69.78%**

TABLE II
EFFECTS OF DIFFERENT EXEMPLAR LISTS (GEMINI-2.5-FLASH)

Selection	# Bins	k-top	Accuracy (%)
randomly selection 1	10	5	39.00
randomly selection 2	10	5	16.47
shortlisting classifier centroids	10	5	8.63

TABLE III
ABLATIONS: EFFECTS OF PRUNING EXEMPLARS (GEMINI-2.5-FLASH)

# Bins	k-top	# Tokens	Accuracy (%)
5	2	0.9K	49.50
5	3	1.0K	42.50
5	4	1.2K	44.50
5	5	1.4K	45.41
10	10	3K+	29.50

TABLE IV
ABLATIONS: EFFECTS OF DISCRETIZING BINS (GEMINI-2.5-FLASH)

# Bins	k-top	Accuracy (%)
5	5	45.41
10	5	39.00
20	5	38.00
30	5	37.00

only 8.63% accuracy, likely because centroidal samples lack the diversity needed for robust in-context learning. Conversely, using a random selection strategy introduced high performance variance, with two separate runs achieving accuracies of 39.00% and 16.47%. This instability underscores the sensitivity of LLMs to the choice of in-context examples and highlights the unreliability of a purely random approach for practical deployment. These findings validate the necessity of a sophisticated and stable exemplar pruning mechanism.

Further ablations confirm the benefits of maintaining a compact prompt structure, as shown in Table III. This experiment analyzes the effect of varying the number of exemplars (k) on accuracy and token count. We observe that increasing k from 4 to 5 provides only a marginal accuracy improvement (from 44.50% to 45.41%) while increasing the prompt length from 1.2K to 1.4K tokens. This indicates diminishing returns beyond a small number of carefully selected examples. More importantly, a significantly larger context, created by setting $k = 10$ and using 10 discretization bins, proves detrimental to performance. In this case, the prompt size balloons to over 3K tokens, and the accuracy drops sharply to 29.50%. This result strongly supports our hypothesis that a concise, focused context is more effective than a large one that may contain distracting or irrelevant information.

Finally, we examine the effect of discretization granularity in Table IV. The results reveal a clear and monotonic trend: coarser quantization consistently leads to better performance in this one-shot, noisy setting. With a fixed $k = 5$, accuracy is highest at 45.41% when using just 5 bins. As the number of bins increases, providing a finer-grained representation of the statistical features, accuracy steadily degrades, dropping to 39.00% with 10 bins and eventually to 37.00% with 30 bins. This suggests that fine-grained distinctions in feature values are less robust to signal noise and that a more abstract, symbolic representation is more conducive to the model's reasoning process.

Exemplar pool construction also affects performance in the shortlisting stage. As shown in Table II, random exemplar pools yield high variance (16.47–39.00%), while a centroid-based pool underperforms (8.63%). This suggests that coverage and diversity of the exemplar set are more critical than proximity to class centroids.

We next study the effect of top-k pruning and token budget (Table III). With 5 discretization bins, increasing k from 4 to 5 yields a small gain (44.50% to 45.41%) while the prompt grows from approximately 1.2K to 1.4K tokens, indicating diminishing returns beyond $k \approx 5$. A larger context with 10 bins and k=10 inflates the prompt to 3K+ tokens and reduces accuracy (29.50%), underscoring the importance of compact contexts.

We finally isolate the effect of the discretization granularity in Table IV. Accuracy degrades monotonically as the number of bins increases (45.41% at 5 bins down to 37.00% at 30 bins), indicating that coarser quantization is more robust for this noisy, one-shot setting.

A. Complexity Analysis

Token Budget. Our framework achieves a substantial reduction in computational cost, primarily through a more efficient use of the token budget. As demonstrated in Table I, the baseline plug-and-play approach requires prompts exceeding 3,000 tokens. In contrast, our method, with its combination of feature discretization and dynamic exemplar pruning, reduces this requirement to between 1.2K and 1.4K tokens (Table III), a decrease of over 50%. This efficiency stems from two key design choices: (i) reducing the number of statistical features from 21 floating-point values to 17 compact symbolic tokens, and (ii) pruning the number of in-context exemplars to a small, targeted set ($k \leq 5$), thereby minimizing redundant information and focusing the model's attention.

Parameter Budget. Beyond token efficiency, our approach enables the use of significantly smaller and more practical LLMs without a prohibitive loss in accuracy. Table I shows that our 5B Gemini 2.5 Flash model achieves an accuracy of 45.41%, which is highly competitive with the 47.80% accuracy of the much larger 32B DeepSeek baseline. This represents an 84% reduction in model parameters, which translates directly to substantially lower VRAM requirements and faster inference speeds. This dramatic reduction in the parameter budget makes in-context AMC feasible for deployment on resource-constrained hardware and edge devices, which is a primary goal of this work.

V. Discussion

Our results demonstrate that with careful prompt engineering, compact LLMs can serve as effective, training-free classifiers for AMC. The core insight of our work is that for in-context learning on noisy, structured data, abstract and concise representations are superior to fine-grained, high-dimensional ones. This ”less is more” principle is evident in our key findings: coarser discretization bins (Table IV) and fewer, more targeted exemplars (Table III) consistently yield better performance. By converting complex floating-point statistics into a small set of symbolic tokens, we not only reduce the token footprint but also provide the LLM with a representation that is more robust to noise and less prone to distraction.

The significant performance gap between our 5B model and the 7B baseline (45.41% vs. 5.20%) underscores that architecture and pre-training are not the only determinants of success; the structure of the prompt is paramount. Our framework achieves a compelling accuracy-efficiency trade-off, making LLM-based AMC practical for real-time applications on constrained hardware. While specialized, supervised models currently lead in absolute accuracy, our approach offers unparalleled flexibility and zero-shot generalization. Future work will focus on closing this performance gap by (i) developing adaptive, data-driven methods for feature selection and binning; (ii) integrating SNR-aware prompting to allow the model to dynamically adjust its reasoning; and (iii) exploring knowledge distillation from larger models like Gemini 2.5 Pro to further enhance the accuracy of the compact Gemini 2.5 Flash model without sacrificing its efficiency.

VI. Conclusion

We introduced Discrete-LLM-AMC, a token- and parameter-efficient framework that makes training-free, in-context automatic modulation classification practical and effective. By discretizing signal statistics into compact symbolic tokens and employing a pruned, targeted prompt structure, we drastically reduce the computational requirements for LLM-based AMC. Our experiments show that this approach cuts prompt length by over 50% and enables a 5B-parameter model to achieve accuracy competitive with a 32B-parameter baseline. These findings demonstrate a viable path toward deploying large language models in resource-constrained wireless communication systems, preserving the benefits of open-set classification while meeting the demands of real-world efficiency.

References

[1] M. Rostami, A. Faysal, R. G. Roshan, H. Wang, N. Muralidhar, and Y.-D. Yao, “Plug-and-play amc: Context is king in training-free, open-set modulation with llms,” *arXiv preprint arXiv:2505.03112*, 2025.

[2] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, “Survey of automatic modulation classification techniques: classical approaches and new trends,” *IET communications*, vol. 1, no. 2, pp. 137–156, 2007.

[3] S. A. Jassim and I. Khider, “Comparison of automatic modulation classification techniques.,” *J. Commun.*, vol. 17, no. 7, pp. 574–580, 2022.

[4] J. Fontaine, A. Shahid, and E. De Poorter, “Towards a wireless physical-layer foundation model: Challenges and strategies,” in *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–7, IEEE, 2024.

[5] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y.-D. Yao, “Modulation classification based on signal constellation diagrams and deep learning,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 3, pp. 718–727, 2018.

[6] T. Huynh-The, C.-H. Hua, Q.-V. Pham, and D.-S. Kim, “Mcnet: An efficient cnn architecture for robust automatic modulation classification,” *IEEE Communications Letters*, vol. 24, no. 4, pp. 811–815, 2020.

[7] A. Faysal, M. Rostami, R. G. Roshan, H. Wang, and N. Muralidhar, “Nmformer: A transformer for noisy modulation classification in wireless communication,” in *2024 33rd Wireless and Optical Communications Conference (WOCC)*, pp. 103–108, IEEE, 2024.

[8] A. Faysal, T. Boushine, M. Rostami, R. G. Roshan, H. Wang, N. Muralidhar, A. Sahoo, and Y.-D. Yao, “Denomae: A multimodal autoencoder for denoising modulation signals,” *IEEE Communications Letters*, 2025.

[9] A. Faysal, M. Rostami, T. Boushine, R. G. Roshan, H. Wang, and N. Muralidhar, “Denomae2. 0: Improving denoising masked autoencoders by classifying local patches,” *arXiv preprint arXiv:2502.18202*, 2025.

[10] H. Ahmadi, S. E. Mahdimahalleh, A. Farahat, and B. Saffari, “Unsupervised time-series signal analysis with autoencoders and vision transformers: A review of architectures and applications,” *arXiv preprint arXiv:2504.16972*, 2025.