LLM AS A CLASSIFIER: LEVERAGING LARGE LANGUAGE MODELS FOR TEXT AND VISION CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Classification is a fundamental capability for AI systems, yet current large language model (LLM) approaches remain poorly suited for latency-critical applications. Prompting and constrained decoding produce verbose, multi-token outputs that require expensive token-by-token generation, while encoder-based models achieve faster inference at the cost of flexibility and generative capacity. We propose LaaC (LLM as a Classifier), a framework that formulates classification as constrained generation with single-token outputs. By introducing atomic label tokens and applying parameter-efficient fine-tuning, our method reduces classification to a deterministic one-step decoding problem. Experiments across text and multimodal benchmarks demonstrate both strong accuracy and consistently fast inference. On MIntRec 2.0, a fine-tuned Gemma-3-27B model attains 62.7% accuracy, outperforming GPT-4o (43.7%) and GPT-5 (51.8%) while running more than an order of magnitude faster. On standard text classification benchmarks, our models match GPT-4o in accuracy while achieving 8× lower tail latency. These results establish decoder-style LLMs as practical and scalable classifiers for real-time applications. Our code is available at `https://anonymous.4open.science/r/LaaC_ICLR`.

1 INTRODUCTION

Classification is a fundamental task in machine learning (Sebastiani, 2002) with widespread applications across domains, from sentiment analysis (Pang et al., 2008) and intent recognition (Goo et al., 2018; Chen et al., 2019) to customer support and interactive dialogue agents. As these applications increasingly operate on multimodal data—combining text and vision—there is growing demand for unified models that can handle diverse input modalities while maintaining efficiency in latency-sensitive environments (Wang et al., 2024).

Current approaches to classification with large language models (LLMs) face significant limitations, particularly in latency-critical applications. Prompt-based methods, while intuitive, often produce verbose, multi-token responses that require additional parsing and introduce substantial inference overhead. More importantly, they provide no guarantee that outputs will be single tokens: a request such as "classify this review as positive or negative" can yield explanatory sentences or multi-token paraphrases rather than clean categorical labels. Even with constrained decoding techniques (Geng et al., 2023) that restrict outputs to valid label strings, models still rely on token-by-token generation. This scales poorly with label vocabulary size and leads to unpredictable latency variations.

This latency challenge is particularly acute in real-world deployment scenarios where classification must occur at scale with strict response time requirements. Traditional encoder-based approaches (e.g., BERT with classification heads) offer predictable, low-latency inference but lack the flexibility and generative capabilities that make modern LLMs attractive for complex reasoning tasks, since they require task-specific architectures and dataset-specific fine-tuning.

In this work, we propose LaaC (LLM as a Classifier), an approach that bridges this gap by treating classification as a constrained generation task with **single-token outputs**. Our key insight is to introduce atomic special tokens (e.g., `[control_1]`) for each class, enabling the model to produce decisions in exactly one generation step. As illustrated in Figure 1, this design not only eliminates

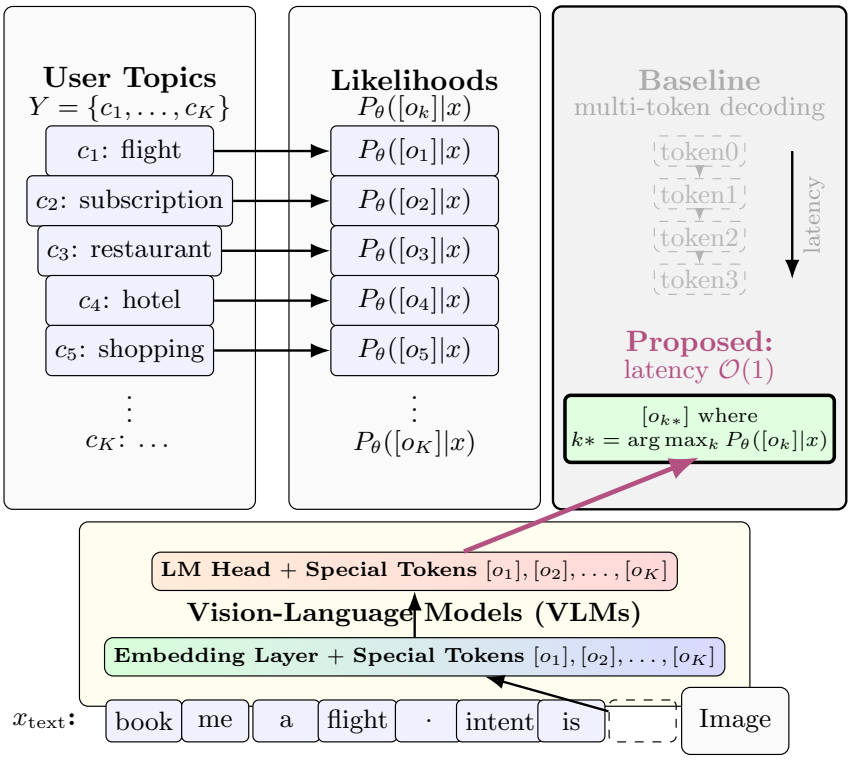


Figure 1: **Overview of our LaaC framework.** Inputs are processed by a decoder-style LLM that directly outputs an atomic special token for the target class. Unlike baseline prompting with multi-token decoding, the framework guarantees $\mathcal{O}(1)$ latency and supports zero-shot adaptation.

multi-token decoding overhead but also enables *zero-shot classification*: by reassigning label tokens at inference time, the model seamlessly adapts to new tasks without task-specific retraining.

We demonstrate this approach using parameter-efficient fine-tuning (LoRA) on vision-language models, creating a unified framework that handles both text-only and multimodal classification tasks. Our method rests on three central pillars: **accuracy**, **latency**, and **generality**. By constraining the output space to a finite set of learned tokens, we achieve deterministic single-step inference while maintaining the semantic understanding capabilities of large pretrained models. Our contributions are threefold:

1. **A unified single-token classification framework** that treats classification as constrained generation, eliminating multi-token decoding inefficiencies while preserving the generalization of decoder-based LLMs.

2. **Significant latency improvements** through atomic label tokens that enable deterministic single-step inference, achieving predictable sub-second inference and maintains efficiency as label spaces grow.

3. **Strong empirical results** across diverse benchmarks, our fine-tuned Gemma-3-27B model outperforms GPT-4o, GPT-5, and encoder-based baselines on multimodal evaluation. On text classification benchmarks, our models remain competitive.

Our approach proves that with careful design, decoder-based models can achieve both the effectiveness and efficiency of specialized encoder architectures while maintaining their broader generative capabilities, making them practical for latency-sensitive applications.

2 Related Work

2.1 Prompt-Based and Few-Shot Classification with LLMs

One natural way to adapt large language models (LLMs) for classification is through *prompting*. For example, a model can be asked: ``Classify the following review as positive or negative: {text}''. While simple, this approach often yields verbose, multi-token answers rather than a clean class symbol. *Few-shot prompting* (Brown et al., 2020) improves reliability by adding in-context demonstrations, and CARP (Sun et al., 2023) further refines this by encouraging models to extract clues and reason step by step. Instruction tuning further enhances this paradigm by aligning models to follow task instructions more robustly (Ouyang et al., 2022; Wei et al., 2021), and multimodal prompting has been demonstrated in vision–language models such as CLIP (Radford et al., 2021). Despite these advances, the outputs of prompting-based methods remain free-form text sequences, which complicates integration into structured applications and hinders efficiency.

To mitigate this, methods such as PET and LM-BFF reformulate classification as a cloze task with *verbalizers* that map each class to a natural-language string and fine-tune the model on small labeled sets (Schick & Schütze, 2020; Gao et al., 2020). Beyond discrete verbalizers, *continuous prompting* methods have been proposed: Prompt Tuning learns soft prompt embeddings optimized for a target task (Ding et al., 2023), while Prefix Tuning prepends trainable key–value vectors to each transformer layer (Li & Liang, 2021). Although effective, these approaches are typically developed for few-shot learning scenarios with limited labeled data; in contrast, our work targets settings with more training examples, where parameter-efficient fine-tuning can be applied to adapt large models.

Recent efforts have also explored *constrained decoding*, where logits are masked or grammar rules are applied so that only valid label strings can be produced (Geng et al., 2023). While this ensures format consistency, it introduces longer latency (due to token-by-token decoding even for short label strings) and limits flexibility when scaling to large label spaces or multimodal tasks. These drawbacks motivate our design of a *finite set of atomic label tokens*, which reduces classification to a single constrained generation step and eliminates the inefficiencies of multi-token verbalizers.

2.2 Encoder-Based Fine-Tuning for Classification

A widely adopted paradigm emphasizes *fine-tuning encoder models* with a dedicated classifier head. Transformer encoders such as BERT and its successors (e.g., RoBERTa, DeBERTa) have become the standard for text classification: the input sequence is processed by the encoder, and the contextualized representation of the [CLS] token is passed through a linear layer trained with cross-entropy loss (Devlin et al., 2019; Liu et al., 2019; He et al., 2020). For multimodal classification, encoder-style fusion models extend this paradigm by incorporating vision or audio encoders and cross-modal attention modules, as in MulT and MAG-BERT (Tsai et al., 2019; Rahman et al., 2020). These approaches are efficient and effective, but they require task-specific classifier heads and lack the flexibility of decoder-based LLMs. In contrast, our method shows that *decoder-style VLMs*, adapted with parameter-efficient fine-tuning (LoRA) and single-token label spaces, can match or surpass encoder baselines on challenging multimodal datasets while maintaining compatibility with generative tasks and supporting a broad range of downstream applications.

3 Methodology

3.1 Problem Formulation

We formalize classification as a supervised learning problem. Each training instance consists of an input tuple

$$x = \left(x^{\text{text}},\, x^{\text{vis}}\right),$$

where x^{text} denotes the textual input and x^{vis} represents the vision modality, which may include static images or short video segments. Not all instances contain both modalities; missing modalities are treated as empty. The objective is to learn a mapping

$$f_\theta : \left(x^{\text{text}},\, x^{\text{vis}}\right) \;\mapsto\; y,$$

where $y \in \mathcal{Y}$ is a categorical label drawn from a predefined set of classes $\mathcal{Y} = \{c_1, c_2, \ldots, c_K\}$.

Given a dataset

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N},$$

the learning goal is to estimate model parameters θ that minimize the expected classification loss:

$$\theta^* = \arg\min_{\theta} \; \mathbb{E}_{(x,y)\sim\mathcal{D}} \; \mathcal{L}\big(f_\theta(x),\, y\big),$$

where $\mathcal{L}$ is the cross-entropy loss over labels. This formulation generalizes unimodal classification to a multimodal setting, where textual and visual inputs are jointly leveraged for improved prediction.

3.2 Model architecture and fine-tuning strategy

Our proposed framework LaaC is model-agnostic and can be applied to a wide range of large language models and vision-language models, including Gemma, Mistral and Qwen architectures. The general principle is to treat classification as a constrained generation task, where the model is guided to produce a single special token corresponding to the target label.

Parameter-Efficient Fine-Tuning. We adopt Low-Rank Adaptation (LoRA) as the primary fine-tuning strategy (Hu et al., 2022). LoRA introduces a small number of trainable rank-decomposition matrices into the attention and projection layers, while keeping the majority of model parameters frozen. This approach enables efficient adaptation across different base models with significantly reduced memory and compute requirements.

Special Tokens for Labels. For each classification category $c_k \in \mathcal{Y}$, we introduce a unique special token $[o_k]$. Conventional choices such as digits or short text labels are problematic: they can appear naturally in the input, causing ambiguity, and larger indices (e.g., "100") are split into multiple subword tokens, leading to multi-step decoding and higher latency. To provide grounding, a natural-language description of each class is included in the system prompt (see Appendix A.2 for prompt templates). During fine-tuning, the model is trained to map $\big(x^{\text{text}}, x^{\text{vis}}\big)$ directly to the correct special token. In implementation, we extend the tokenizer with 500 reserved control tokens $[o_1], \ldots, [o_{500}]$, which allows the model to accommodate up to 500 classes during inference without retraining.

Training Objective. The training objective is defined as cross-entropy loss over the predicted control token. Since classification requires only a single output, the loss is computed exclusively on the final assistant response token (e.g., $[o_k]$), while all preceding tokens in the input sequence are masked out and excluded from loss calculation. Formally, for each instance (x, k), the loss is

$$\mathcal{L} = -\log P_\theta\big([o_k] \,\big|\, x\big).$$

3.3 Special Token Design (Single-Token Outputs)

Let $\mathcal{Y} = \{c_1, \ldots, c_K\}$ denote the label set and let $\mathcal{V}$ be the base tokenizer vocabulary. We augment the vocabulary with a set of K *atomic output symbols*:

$$\Omega = \{[o_1], [o_2], \ldots, [o_K]\}, \qquad \tilde{\mathcal{V}} = \mathcal{V} \cup \Omega.$$

Each $[o_k]$ is a dedicated *control token* corresponding to class c_k, with a trainable embedding $e_k \in \mathbb{R}^d$. These rows are appended to the model's embedding matrix and are updated during fine-tuning, together with LoRA-adapted weights.

Why single-token outputs. If labels are represented as natural-language strings s_k (e.g., `"book a flight"`), a subword tokenizer typically produces a variable-length sequence

$$\tau(s_k) = (v_1, \ldots, v_{m_k}), \quad m_k \geq 1,$$

where $\tau(\cdot)$ denotes the tokenization function. This variability requires loss over multiple decoding steps and depends on segmentation. In contrast, our design assigns each class a single atomic symbol $\omega_k = [o_k] \in \Omega$, which is guaranteed to decode as exactly one token. This yields three benefits: (i) the output space collapses to K symbols, (ii) ambiguity from label strings that may appear in the input is eliminated, and (iii) decoding and evaluation are simplified to a deterministic one-step classification.

Randomized label assignments. To prevent memorization of static token–label associations, we do not fix a permanent mapping between special tokens and semantic classes. Instead, during preprocessing we randomly shuffle the correspondence between classes and control tokens across training instances. This design encourages the model to rely on the contextual descriptions of labels provided in the prompt, rather than memorizing token identities. As a result, the model learns to infer the correct output token from the input context, which improves robustness and generalization across datasets and label spaces. Without randomization, models tended to overfit to token IDs and failed to generalize to new mappings.

Separation from context. We reserve a sentinel namespace for Ω, ensuring that these tokens are never decomposed into subwords and never occur in the input text. Formally, $\Omega \cap \mathcal{V} = \emptyset$ and $\Omega \cap \tau(x) = \emptyset$ for any input x. In practice, this is enforced by registering $[o_k]$ as special tokens in the tokenizer, so that they are available to the model during output generation.

Training objective (single position). Given input $x = \left(x^{\text{text}}, x^{\text{vis}}\right)$ with gold label $k \in \{1, \ldots, K\}$, the model is required to emit exactly one control token $[o_k]$. Let h_T denote the decoder state at the output position (the single assistant step). We compute logits restricted to Ω:

$$z_j = (W h_T + b)_j, \qquad P_\theta([o_j] \mid x) = \frac{\exp(z_j)}{\sum_{i=1}^{K} \exp(z_i)} \quad \text{for } j \in \{1, \ldots, K\}.$$

The loss is standard cross-entropy over this single prediction:

$$\mathcal{L}(x, k) = -\log P_\theta([o_k] \mid x).$$

All preceding tokens (system/user prompts and any in-context descriptions) are masked out of the loss, so classification supervision is concentrated solely on the final output position. In implementation, we construct a binary loss mask: the position of the response token is assigned its class label, while all other positions are set to -100, which the cross-entropy loss ignores.

Inference rule. At test time, decoding reduces to a single restricted argmax:

$$k^* = \arg \max_{k \in \{1, \ldots, K\}} P_\theta([o_k] \mid x).$$

Equivalently, we set `max_new_tokens = 1` and restrict the decision space to Ω, ensuring the model outputs exactly one control token (e.g., $[o_{k^*}]$) rather than a multi-token string. This deterministic procedure avoids token-by-token generation and guarantees constant-time inference.

4 Experiments

4.1 Datasets

Training corpus. We construct a balanced training corpus of 28k examples, comprising 14k text-only and 14k multimodal classification instances. **Text datasets.** The text-only portion includes examples, each consisting of a natural language input paired with a categorical label. The corpus spans diverse domains, specifically, we use the CLINC dataset (Larson et al., 2019) (150 intents across 10 domains, with out-of-scope examples), the AMZN-MASSIVE dataset (FitzGerald et al., 2022; Bastianelli et al., 2020) (60 intents spanning 18 domains), and MULTIWOZ-2.2 (Zang et al., 2020) (11 intents across 3 domains). To encourage generalization beyond standard datasets, we additionally incorporate constrained instruction-following data from FollowBench and Unnatural Instructions (Jiang et al., 2023; Honovich et al., 2022), as well as synthetic data generated with AgentGym (Xi et al., 2024). **Multimodal datasets.** The multimodal portion comprises 14k examples from established vision–language datasets. We include video–text pairs from MINTREC (Zhang et al., 2022) and reformulate image–question pairs from subsets of A-OKVQA (Schwenk et al., 2022) and VISUAL7W (Zhu et al., 2016). To construct this corpus, we sample approximately 5k examples each from the image-based datasets while retaining all available MINTREC frames. All data are reformulated into a unified JSON schema (see Fig. 2), with consistent `messages` fields and an optional `image_path`. To mitigate label memorization, we randomize control-token assignments: class labels are mapped to tokens drawn from a reserved set of 500 control tokens.

Example of a Classification Training Instance

```
{
"messages": [
  { "role": "system", "content": "You are a classification
expert. Topics: [control_x] Complain, [control_y] Praise,
[control_z] Apologise ..." },
  { "role": "user", "content": "### USER CONVERSATION HERE ###" },
  { "role": "assistant", "content": "[control_x]" }
],
"image_path": "path/to/image.jpg",
}
```

Figure 2: Illustration of a fine-tuning training instance in our classification datasets. Each sample includes the structured `messages` field and optional `image_path`.

Evaluation benchmarks. We evaluate our approach on a diverse suite of multimodal and text-only classification benchmarks. For multimodal evaluation, we use the official test split of **MIntRec2.0** (Zhang et al., 2024), a large-scale benchmark for intent recognition in multimodal dialogues that combine text and vision. This dataset includes 30 fine-grained intent classes and requires reasoning over conversational context and multiple modalities, making it a challenging test of real-world multimodal classification. For text-only tasks, we consider four widely used benchmarks: **SST-2** (Socher et al., 2013) is a binary sentiment analysis benchmark consisting of movie reviews annotated as positive or negative; **Amazon Reviews Polarity** (McAuley & Leskovec, 2013; Zhang et al., 2015) contains millions of product reviews labeled as positive or negative. It evaluates sentiment classification in a large-scale, noisy e-commerce domain; **AG News** (Zhang et al., 2015) is a topic classification dataset with four categories: World, Sports, Business, and Science/Technology; and **DBpedia** (Lehmann et al., 2015) is a 14-class benchmark built from Wikipedia articles, covering categories such as Company, Artist, Athlete, and Place. It provides a broad test of factual and encyclopedic text classification. Example prompt templates for these datasets are provided in Appendix A.2 (Figures 4 to 8). For text-only benchmarks, we evaluate on 200 randomly sampled test examples from each dataset to ensure consistent and efficient comparisons across models.

4.2 Baselines

We benchmark against both open-source and proprietary systems.

Pretrained LLMs. We include the untuned versions of Gemma-3 (4B, 27B) and Mistral-3-24B as base VLM checkpoints. These serve as reference points for the capability of large pretrained models without classification adaptation.

External API models. For stronger upper-bound comparisons, we evaluate proprietary multimodal models including GPT-4o and GPT-5 (including GPT-5-nano). These systems represent state-of-the-art commercial offerings.

Encoder-based models. To contextualize against specialized architectures, we also report results for strong encoder-style multimodal baselines (e.g., `MAG-BERT` and `MulT`), following the MIntRec 2.0 benchmark protocol.

4.3 Training Details

We fine-tuned both Gemma-3 and Mistral-3 with a batch size of 1 and gradient accumulation of 16 (effective batch size 16). Models were trained for 30 epochs using a learning rate of 2×10^{-5} with a warmup ratio of 0.1. LoRA modules (rank 8, $\alpha = 16$, dropout 0.05) were applied to attention and feed-forward layers. Gemma-3 employed tied embeddings, whereas Mistral-3 used untied embeddings that required explicit saving. Training was conducted on 8×NVIDIA A100 GPUs with mixed precision (bfloat16), gradient checkpointing, and DeepSpeed ZeRO-3. Early stopping was applied with a patience of 8 validation steps and a minimum improvement threshold of 10^{-4}. Model performance was evaluated every 500 steps using validation loss as the criterion.

4.4 EVALUATION METRICS

We evaluate models along two complementary dimensions: classification effectiveness and inference efficiency. **Classification accuracy.** The primary metric is accuracy, computed as the percentage of instances where the predicted control token matches the gold label. Accuracy is reported separately for each dataset. **Latency.** To capture efficiency, we measure model response times. We report the median latency (P50) and the tail latency (P95) across evaluation batches. All baselines are evaluated with our vLLM-based inference framework (Kwon et al., 2023) on a single NVIDIA A100 GPU, using consistent input formatting and datasets.

4.5 RESULTS

4.5.1 MULTIMODAL EVALUATION RESULTS

We evaluate our fine-tuned models trained on the multimodal portion of our corpus on the challenging MIntRec 2.0 dataset, which requires understanding multimodal dialogue contexts (text, image, video) for intent recognition. The evaluation covers both base models and fine-tuned variants of Gemma-3 and Mistral-3, with comparisons against GPT models such as GPT-4o and GPT-5, as well as encoder-based baselines reported in the original paper.

MIntRec 2.0 (Multimodal Topic Classification). Table 1 presents results on MIntRec 2.0. Base Gemma-3 models perform poorly ($\sim$16–18% accuracy), underscoring the difficulty of multimodal intent recognition without adaptation. After fine-tuning, performance improves markedly: Gemma-3-4B reaches **55.2%**, while Gemma-3-27B achieves **62.7%**, significantly outperforming GPT-4o (43.7%) and GPT-5 (51.8%) despite being much smaller and faster. Fine-tuned Gemma-3 models also achieve low latency (P95 $<$ 1s), in contrast to GPT-4o and GPT-5, which require 6–13s. This efficiency makes our fine-tuned models practical for real-time applications.

Comparison with Encoder-Based Models. MAG-BERT and MulT, strong encoder-based multimodal baselines from the original MIntRec paper, achieve 60.6% accuracy. Our fine-tuned Gemma-3-27B surpasses both, reaching **62.7%**, while also offering the benefits of a unified generative modeling framework. This result is notable since encoder-based approaches are specifically optimized for multimodal classification with carefully designed fusion architectures, whereas our approach adapts a general-purpose generative model. The competitive or superior performance of fine-tuned LLMs indicates that large decoder-based architectures, when fine-tuned on in-domain data, can match or outperform specialized encoder-based models while simultaneously enabling broader generative and reasoning capabilities.

Table 1: MIntRec 2.0 evaluation results, including baselines from the original paper ($^\diamond$) and our evaluations. FT = fine-tuned. Models are sorted by accuracy. Speedup is relative to GPT-4o (P50 = 4.30s, P95 = 6.12s).

Model	Accuracy (%)	P50 (s)	Speedup (P50)	P95 (s)	Speedup (P95)
Gemma-3-4B (Base)	16.04	1.33	3.23×	1.77	3.46×
Gemma-3-27B (Base)	17.76	2.18	1.97×	2.77	2.21×
Mistral-3-24B (Base)	40.83	0.77	5.58×	1.71	3.58×
GPT-5-nano	41.02	3.67	1.17×	5.46	1.12×
GPT-4o	43.68	4.30	1.00×	6.12	1.00×
Mistral-3-24B (FT, LaaC)	**49.34**	**0.64**	**6.72×**	**1.64**	**3.73×**
GPT-5	51.84	7.13	0.60×	13.01	0.47×
Gemma-3-4B (FT, LaaC)	**55.19**	**0.26**	**16.54×**	**0.60**	**10.20×**
MAG-BERT$^\diamond$	60.58	–	–	–	–
MulT$^\diamond$	60.66	–	–	–	–
Gemma-3-27B (FT, LaaC)	**62.72**	**0.37**	**11.62×**	**0.90**	**6.80×**

4.5.2 TEXT CLASSIFICATION ACROSS DOMAINS

Our evaluations span four widely used classification benchmarks: SST-2, Amazon Reviews, AG News, and DBpedia. Unlike encoder-based baselines (e.g., BERT or RoBERTa variants), which are

Table 2: Text-only evaluation results on SST-2, Amazon Reviews, AG News, and DBpedia. We report accuracy and latency (median *P50* and tail *P95*).

Dataset	Model	Acc. (%)	P50 Latency (s)	P95 Latency (s)
SST-2	GPT-4o	95.50	0.53	0.84
	Mistral-3-24B (Base)	95.50	0.07	0.07
	Mistral-3-24B (FT, LaaC)	95.50	**0.03**	**0.03**
	Gemma-3-27B (Base)	95.00	0.36	0.41
	Gemma-3-27B (FT, LaaC)	95.50	**0.11**	**0.12**
Amazon Reviews	GPT-4o	95.00	0.53	0.97
	Mistral-3-24B (Base)	95.50	0.08	0.09
	Mistral-3-24B (FT, LaaC)	95.00	**0.05**	**0.08**
	Gemma-3-27B (Base)	93.50	0.38	0.46
	Gemma-3-27B (FT, LaaC)	94.00	**0.10**	**0.12**
AG News	GPT-4o	84.50	0.55	1.00
	Mistral-3-24B (Base)	84.00	0.07	0.17
	Mistral-3-24B (FT, LaaC)	83.00	**0.05**	**0.05**
	Gemma-3-27B (Base)	84.00	0.25	0.60
	Gemma-3-27B (FT, LaaC)	81.50	**0.11**	**0.13**
DBpedia	GPT-4o	97.00	0.61	1.23
	Mistral-3-24B (Base)	94.50	0.10	0.21
	Mistral-3-24B (FT, LaaC)	93.00	**0.05**	**0.06**
	Gemma-3-27B (Base)	97.00	0.25	0.48
	Gemma-3-27B (FT, LaaC)	95.00	**0.10**	**0.11**

typically fine-tuned directly on each dataset and thus achieve strong but task-specific results, our framework is *not fine-tuned on any of these datasets*. Instead, we evaluate zero-shot generalization by comparing our fine-tuned classifier against its untuned base model and GPT-4o.

Without task-specific adaptation, LaaC demonstrates **robust cross-domain performance**. As shown in Table 2, on sentiment classification (SST-2, Amazon Reviews) and encyclopedic categorization (DBpedia), our fine-tuned models achieve accuracy comparable to GPT-4o and close to its base model, while maintaining consistently sub-200 ms inference latency (P95 $\leq$ 0.13 s).

Most notably, the efficiency gains are substantial. While GPT-4o requires close to one second for tail latency on these tasks, our approach reduces this by nearly an order of magnitude. By collapsing classification into a deterministic single-token decision, inference time becomes both **fast and predictable**, which is crucial for latency-sensitive deployments.

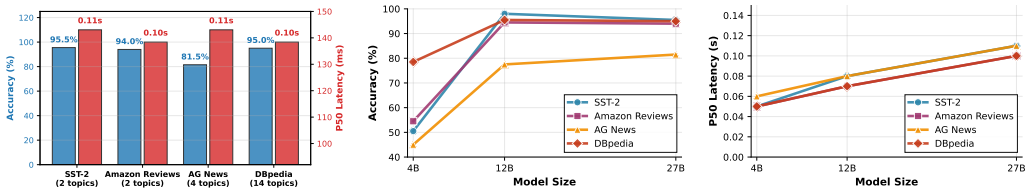
Overall, these results highlight that our approach **generalizes well across unseen text domains without any task-specific fine-tuning**. In contrast to encoder models that trade flexibility for efficiency, our method preserves the generative capacity of large decoder LLMs while matching their classification accuracy and surpassing them in efficiency.

4.6 Effect of label-set sizes

We further analyze the impact of the number of label sets on model performance by evaluating across datasets with increasing topic sizes, ranging from binary sentiment classification (SST-2, Amazon Reviews) to multi-class categorization (AG News with 4 topics and DBpedia with 14 topics).

Generalization ability. As shown in Figure 3a, our fine-tuned Gemma-3-27B model consistently maintains high accuracy across datasets of varying difficulty. Even as the label space expands from 2 to 14 categories, the accuracy remains 95% on DBpedia and comparable to binary sentiment datasets, demonstrating strong zero-shot generalization ability.

Efficiency stability. In addition to accuracy, we examine efficiency via P50 latency. The results reveal that latency remains remarkably stable across datasets, fluctuating only within 0.10–0.11s despite the growth in label space. This indicates that our design achieves scalable inference efficiency while handling tasks of increasing complexity.



(a) Topic cardinality. (b) Accuracy vs model size. (c) Latency vs model size.

Figure 3: Performance of Gemma-3 models across datasets.

4.7 Scaling analysis

We further investigate how accuracy and efficiency scale with model sizes. Figures 3b and 3c report results for Gemma-3 models with 4B, 12B, and 27B parameters on four text benchmarks. **Accuracy.** Performance improves consistently as model size increases (Figure 3b). The 4B model shows lower accuracy, particularly on AG News, while the 12B model closes much of the gap. The 27B model achieves the strongest results across all datasets, exceeding 95% on SST-2 and DBpedia and remaining above 94% on Amazon Reviews. **Latency.** Inference latency remains nearly constant across scales (Figure 3c). Median (P50) latencies vary only between 0.02–0.06s, despite the 7× parameter increase from 4B to 27B. This stability stems from reducing classification to a single-token decision, which eliminates the usual token-by-token decoding overhead. These results demonstrate that larger models yield predictable accuracy gains while efficiency remains effectively unchanged.

4.8 Effect of text training data

To assess the value of incorporating both unimodal and multimodal supervision, we evaluated fine-tuned Gemma-3-27B on a held-out proprietary dataset. When trained only on multimodal data, the model achieved 80.6% accuracy with P50 = 0.06s and P95 = 0.18s. By contrast, when training combined both text-only and multimodal data, accuracy improved to 84.7% while maintaining comparable efficiency (P50 = 0.06s, P95 = 0.06s). These results suggest that exposing the model to both text and multimodal supervision during training provides stronger representations and leads to consistent accuracy gains. A similar improvement (≈0.5%) at unchanged latency was also observed on DBpedia, indicating that the effect generalizes beyond a single dataset.

5 Conclusion

We introduced a framework LaaC that treats classification as a constrained generation problem with *single-token* outputs. By augmenting decoder-style LLMs with atomic label tokens and adapting them through parameter-efficient fine-tuning, our method collapses classification into a deterministic one-step decoding task. This design achieves significant latency reductions while preserving the generative flexibility of large models. Empirically, fine-tuned Gemma-3 models outperform much larger proprietary systems on the MIntRec 2.0 benchmark and match or surpass encoder-based multimodal baselines, all while running with sub-second tail latency. These results demonstrate that decoder LLMs can serve as practical, scalable classifiers for latency-sensitive applications.

Our current study focuses on text and vision inputs; extending the framework to additional modalities such as audio remains an open direction. While our evaluation highlights substantial latency improvements, a deeper analysis of calibration, robustness, and multilingual generalization is needed to validate deployment readiness. Future work will explore scaling to larger label spaces, integrating rejection mechanisms for out-of-scope detection, and combining single-token classification with reasoning-augmented LLMs. Together, these directions aim to advance LLMs as both versatile generators and efficient classifiers for real-world multimodal systems.

References

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. SLURP: A spoken language understanding resource package. In *Proceedings of the 2020 Conference on Em-*

pirical Methods in Natural Language Processing (EMNLP), pp. 7252–7262, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.588. URL https://aclanthology.org/2020.emnlp-main.588.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Qian Chen, Zhu Zhuo, and Wen Wang. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235, 2023.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages, 2022.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured nlp tasks without finetuning. *arXiv preprint arXiv:2305.13971*, 2023.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 753–757, 2018.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor, 2022. URL https://arxiv.org/abs/2212.09689.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. Followbench: A multi-level fine-grained constraints following benchmark for large language models, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. URL https://www.aclweb.org/anthology/D19-1131.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172, 2013.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2):1–135, 2008.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Wasifur Rahman, Md Kamrul Hasan, Sangwu Lee, Amir Zadeh, Chengfeng Mao, Louis-Philippe Morency, and Ehsan Hoque. Integrating multimodal information in large pretrained transformers. In *Proceedings of the conference. Association for computational linguistics. Meeting*, volume 2020, pp. 2359, 2020.

Timo Schick and Hinrich Schütze. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*, 2020.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pp. 146–162. Springer, 2022.

Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. Text classification via large language models. *arXiv preprint arXiv:2305.08377*, 2023.

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the conference. Association for computational linguistics. Meeting*, volume 2019, pp. 6558, 2019.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. Agentgym: Evolving large language model-based agents across diverse environments, 2024.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, ACL 2020*, pp. 109–117, 2020.

Hanlei Zhang, Hua Xu, Xin Wang, Qianrui Zhou, Shaojie Zhao, and Jiayan Teng. Mintrec: A new dataset for multimodal intent recognition. In *Proceedings of the 30th ACM international conference on multimedia*, pp. 1688–1697, 2022.

Hanlei Zhang, Xin Wang, Hua Xu, Qianrui Zhou, Kai Gao, Jianhua Su, Wenrui Li, Yanting Chen, et al. Mintrec2. 0: A large-scale benchmark dataset for multimodal intent recognition and out-of-scope detection in conversations. *arXiv preprint arXiv:2403.10943*, 2024.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4995–5004, 2016.

A Appendix

A.1 LLM Usage Disclosure

We used large language models (e.g., ChatGPT) only to polish writing and assist with literature search. They were not used for generating research ideas or results.

A.2 Example prompts.

Example Prompt (SST-2)

```
{
"messages": [
  { "role": "system", "content": "You are a classification
expert. Topics: [control_1] Negative, [control_2] Positive. Based
on the overall sentiment expressed in this review, respond with the
relevant control token:" },
  { "role": "user", "content": "it's a charming and often
affecting journey." },
  { "role": "assistant", "content": "[control_2]" }
],
"image_path": "none",
}
```

Figure 4: Prompt template for classifying SST-2 movie reviews.

Example Prompt (AG News)

```
{
"messages": [
  { "role": "system", "content": "You are a classification
expert. Topics: [control_1] World, [control_2] Sports, [control_3]
Business, [control_4] Science/Technology. Based on the content of
this article, respond with the relevant control token:" },
  { "role": "user", "content": "Fears for T N pension after
talks. Unions representing workers at Turner Newall say they are
'disappointed' after talks with stricken parent firm Federal
Mogul." },
  { "role": "assistant", "content": "[control_3]" }
],
"image_path": "none",
}
```

Figure 5: Illustration of an inference prompt from AG News (Business category).

Example Prompt (Amazon Reviews)

```
{
"messages": [
  { "role": "system", "content": "You are a classification
expert. Topics: [control_1] Negative, [control_2] Positive. Based
on the overall sentiment expressed in this review, respond with the
relevant control token:" },
  { "role": "user", "content": "DVD Player crapped out after one
year. I also began having the incorrect disc problems that I've
read about on here. The VCR still works, but the DVD side is
useless. I understand that DVD players sometimes just quit on you,
but after not even one year? To me that's a sign of bad quality.
I'm giving up JVC after this as well. I'm sticking to Sony or
giving another brand a shot." },
  { "role": "assistant", "content": "[control_1]" }
],
"image_path": "none",
}
```

Figure 6: Illustration of an inference prompt from Amazon Reviews (Negative sentiment).

Example Prompt (DBpedia)


```
{
"messages": [
  { "role": "system", "content": "You are a classification
expert. Topics: [control_1] Company, [control_2]
EducationalInstitution, [control_3] Artist, [control_4] Athlete,
[control_5] OfficeHolder, [control_6] MeanOfTransportation,
[control_7] Building, [control_8] NaturalPlace, [control_9] Village,
[control_10] Animal, [control_11] Plant, [control_12] Album,
[control_13] Film, [control_14] WrittenWork. Based on the content
of this article, respond with the relevant control token:" },
  { "role": "user", "content": "Pizza Port Brewing Company is a
brewpub with five locations in Southern California: Solana Beach,
two in Carlsbad (Downtown and Bressi Ranch), Ocean Beach and San
Clemente. A former Pizza Port location in San Marcos spun out of
Pizza Port in 2006 and is now an independent operation, the Port
Brewing Company / Lost Abbey brewery. It has received multiple
awards, including S̈mall Brewpub of the Yearf̈or both 2003 and 2004
by the Great American Beer Festival and six awards for its beers at
the World Beer Cup." },
  { "role": "assistant", "content": "[control_1]" }
],
"image_path": "none",
}
```


Figure 7: Illustration of an inference prompt from DBpedia (Company category).

Example of a Classification Training Instance (Topic Classification)


```
{
"messages": [
  { "role": "system", "content": "You are a topic classification
expert. Before making a decision, carefully follow all the
topic-specific instructions/descriptions. Topics: [control_1]
Acknowledge, [control_2] Advise, [control_3] Agree, [control_4]
Apologise, [control_5] Arrange, [control_6] Ask for help, [control_7]
Asking for opinions, [control_8] Care, [control_9] Comfort,
[control_10] Complain, [control_11] Confirm, [control_12] Criticize,
[control_13] Doubt, [control_14] Emphasize, [control_15] Explain,
[control_16] Flaunt, [control_17] Greet, [control_18] Inform,
[control_19] Introduce, [control_20] Invite, [control_21] Joke,
[control_22] Leave, [control_23] Oppose, [control_24] Plan,
[control_25] Praise, [control_26] Prevent, [control_27] Refuse,
[control_28] Taunt, [control_29] Thank, [control_30] Warn. Based on
the above conversation, respond with the relevant topic ID:" },
  { "role": "user", "content": "Thank you so much for your help!
I really appreciate it." },
  { "role": "assistant", "content": "[control_29]" }
],
"image_path": "none",
}
```


Figure 8: Illustration of a prompt from the Topic Classification dataset (Thank intent).