

NanoLLaDA

Fine-Tuning and Evaluation of a
Diffusion-Based Language Model

MAY 2025



Pupipat
Singkhorn



Chanatip
Pattanapen



Kawin
Rattanapun



Chotpisit
Adunsehawat

arXiv:2502.09992v2 [cs.CL] 18 Feb 2025

Large Language Diffusion Models

Shen Nie^{1*†} Fengqi Zhu^{1*†} Zebin You^{1†} Xiaolu Zhang^{2‡} Jingyang Ou¹ Jun Hu^{2‡} Jun Zhou²
Yankai Lin^{1‡} Ji-Rong Wen¹ Chongxuan Li^{1†‡}

Abstract

Autoregressive models (ARMs) are widely regarded as the cornerstone of large language models (LLMs). We challenge this notion by introducing LLaDA, a diffusion model trained from scratch under the pre-training and supervised fine-tuning (SFT) paradigm. LLaDA models distributions through a forward data masking process and a reverse process, parameterized by a vanilla Transformer to predict masked tokens. By optimizing a likelihood bound, it provides a principled generative approach for probabilistic inference. Across extensive benchmarks, LLaDA demonstrates strong *scalability*, outperforming our self-constructed ARM baselines. Remarkably, LLaDA 8B is competitive with strong LLMs like LLaMA3 8B in *in-context learning* and, after SFT, exhibits impressive *instruction-following* abilities in case studies such as multi-turn dialogue. Moreover, LLaDA addresses the reversal curse, surpassing GPT-4o in a reversal poem completion task. Our findings establish diffusion models as viable and promising alternative to ARMs, challenging the assumption that key LLM capabilities discussed above are inherently tied to ARMs. Project page and codes: <https://ml-gsai.github.io/LLaDA-demo/>.

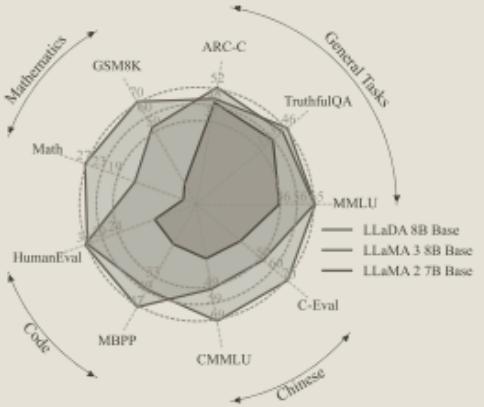


Figure 1. Zero/Few-Shot Benchmarks. We scale LLaDA to an unprecedented size of 8B parameters from scratch, achieving competitive performance with strong LLMs (Dubey et al., 2024).

distribution $p_{\text{data}}(\cdot)$ by optimizing a model distribution $p_{\theta}(\cdot)$ through maximum likelihood estimation, or equivalently KL divergence minimization between the two distributions:

$$\max_{\theta} \mathbb{E}_{p_{\text{data}}(x)} \log p_{\theta}(x) \Leftrightarrow \min_{\theta} \underbrace{\text{KL}(p_{\text{data}}(x) || p_{\theta}(x))}_{\text{Generative modeling principles}}. \quad (1)$$

1. Introduction

What is now proved was once only imagined.
—William Blake

Large language models (LLMs) (Zhao et al., 2023) fall entirely within the framework of *generative modeling*. Specifically, LLMs aim to capture the true but unknown language

*Equal contribution †Work done during an internship at Ant Group [‡]Project leaders ¹Gaoling School of Artificial Intelligence, Renmin University of China; Beijing Key Laboratory of Big Data Management and Analysis Methods ²Ant Group. *Correspondence to: Chongxuan Li <chongxuanli@ruc.edu.cn>.

Preprint.

The predominant approach relies on the *autoregressive modeling* (ARM)—commonly referred to as the *next-token prediction* paradigm—to define the model distribution:

$$p_{\theta}(x) = p_{\theta}(x^1) \prod_{i=2}^L p_{\theta}(x^i | x^1, \dots, x^{i-1}), \quad (2)$$

where x is a sequence of length L , and x^i is the i -th token.

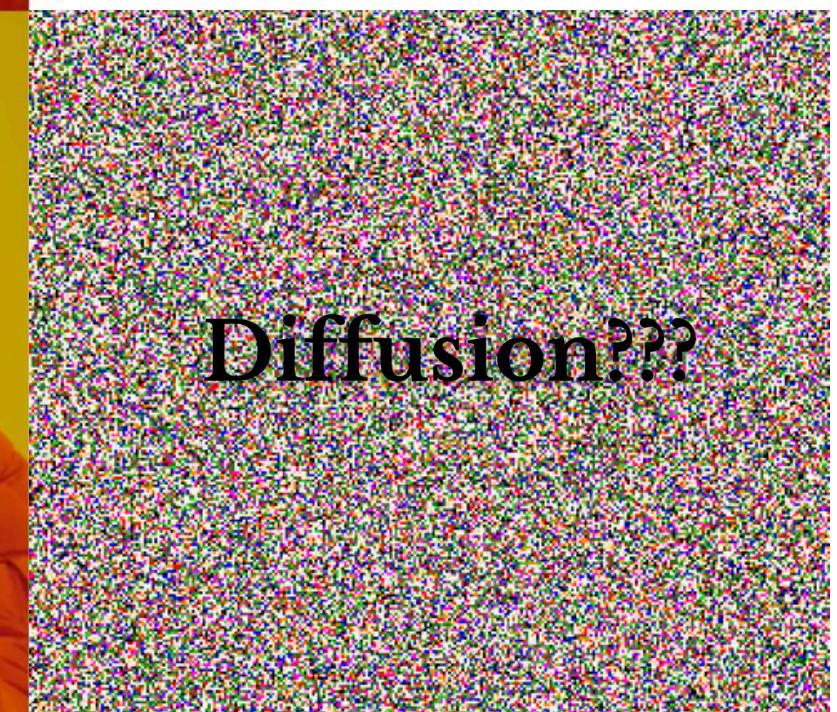
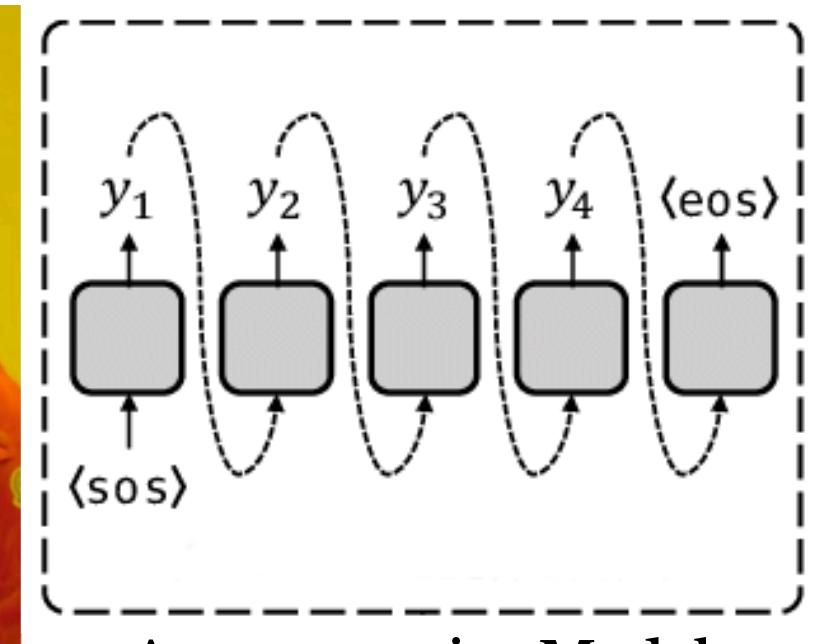
This paradigm has proven remarkably effective (Radford, 2018; Radford et al., 2019; Brown, 2020; OpenAI, 2022) and has become the foundation of current LLMs. Despite its widespread adoption, a fundamental question remains unanswered: *Is the autoregressive paradigm the only viable path to achieving the intelligence exhibited by LLMs?*

Motivating Work: Diffusion as an Alternative to Autoregression

Our work is inspired by
"Large Language Diffusion Models" by Nie et al. (2025)
arXiv:2502.09992

LLaDA - Large **L**anguage **D**iffusion with m**A**sking

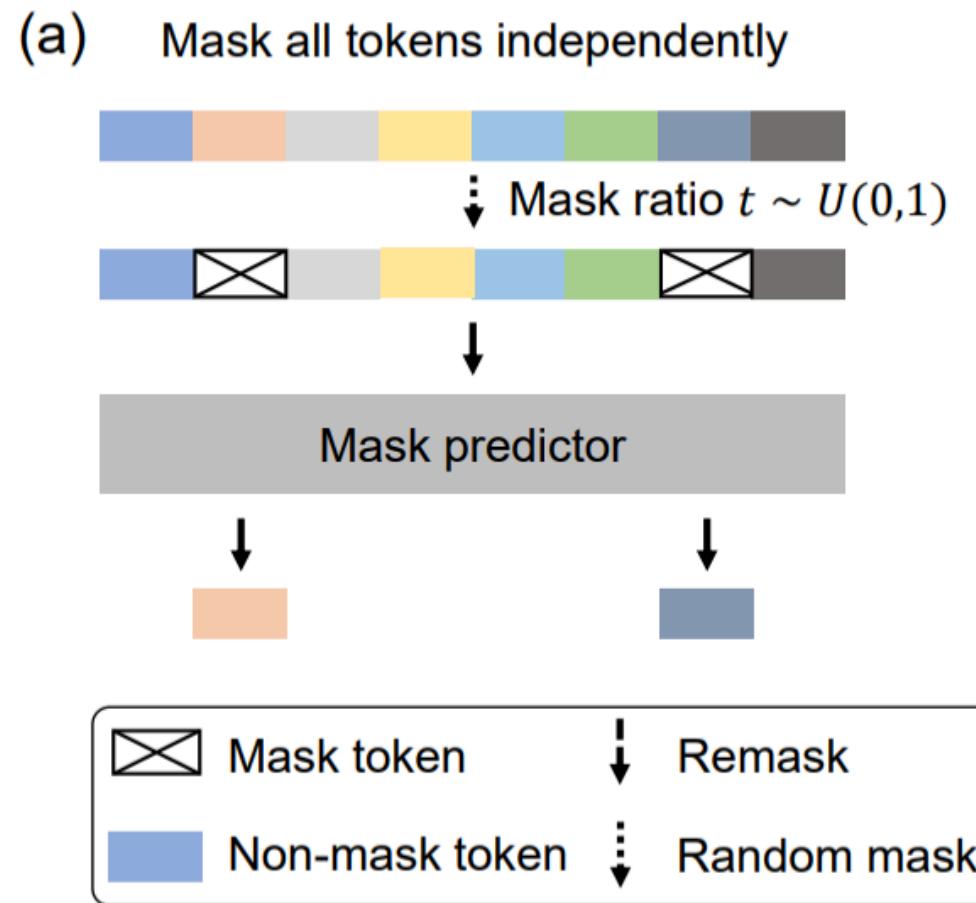
- Diffusion-based alternative to AR models
- Masked reverse denoising process
- Scalable, bidirectional generation
- Supports multilingual input, including Thai



LLaDA Architecture and Methodology

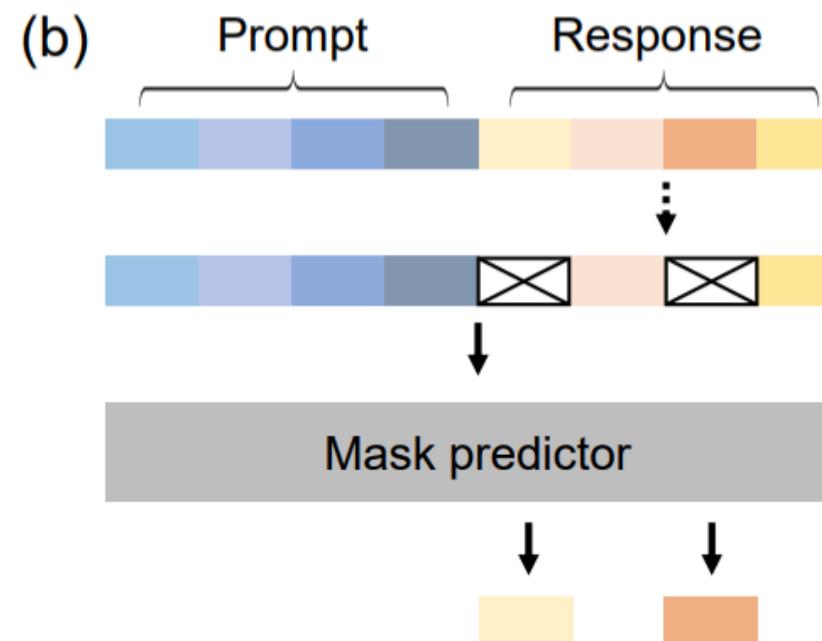
Pre-training

- 2.3 trillion tokens from high-quality, filtered web data includes code, math, and multilingual text
- Transformer encoder-style mask predictor



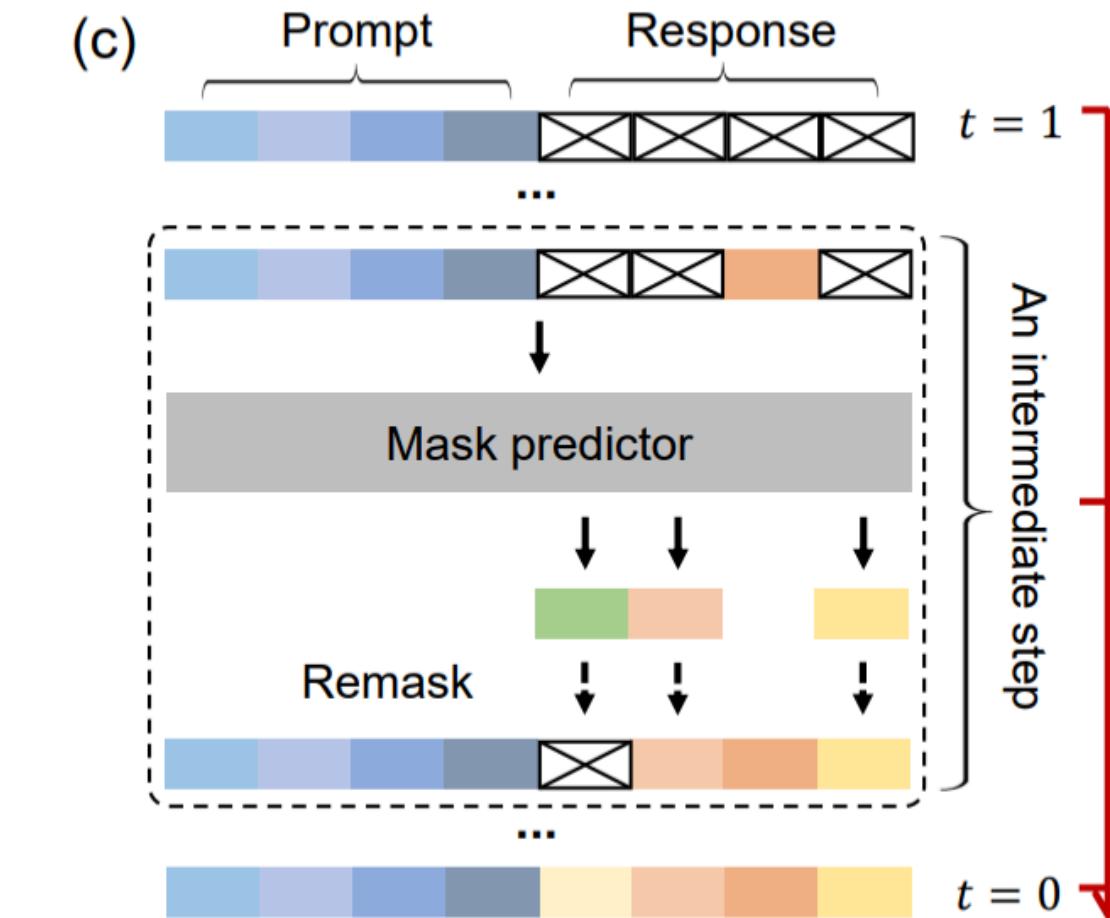
Supervised Fine-Tuning (SFT)

- 4.5 million prompt–response pairs
- No Reinforcement Learning (RLHF), no instruction alignment tricks



Inference

- Iterative reverse denoising process
 - Predict all masked tokens
 - Remask a subset (randomly or based on low-confidence)
- Non-autoregressive sampling
- Parallel token prediction



LLaDA Performance Across Benchmarks

Zero-/Few-Shot Performance

Comparable to LLaMA2-7B/3-8B on MMLU, GSM8K, HumanEval -

Stronger results in **mathematics** and **Chinese** benchmarks -
Relatively lower scores on **code generation** tasks -

Instruction-Following

Notable improvement after supervised fine-tuning (SFT) -
Competitive in **multi-turn reasoning** without RL alignment -

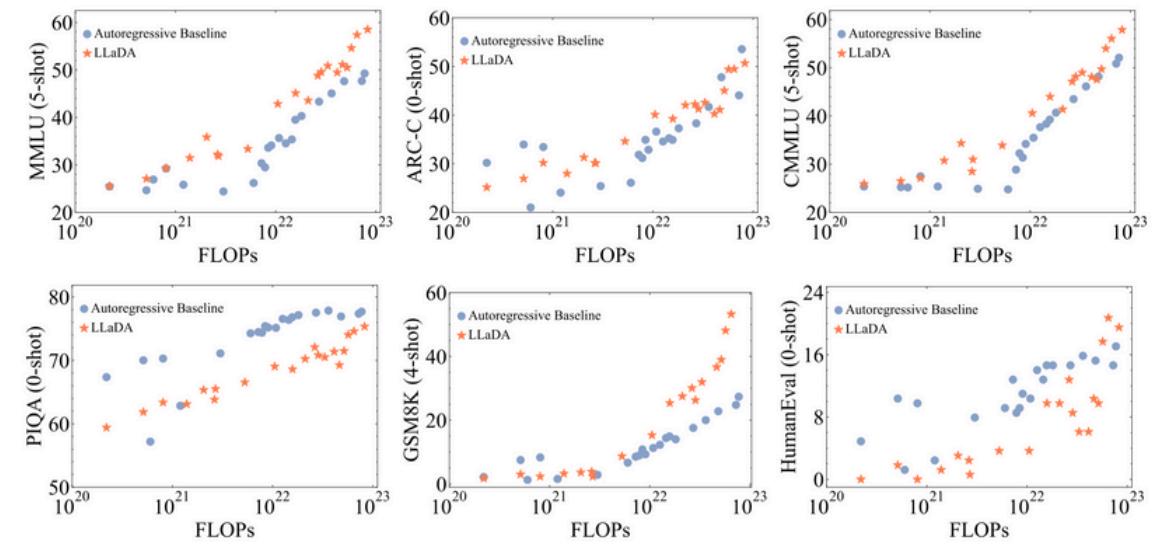


Figure 3. **Scalability of LLaDA.** We evaluate the performance of LLaDA and our ARM baselines trained on the same data across increasing computational FLOPs. LLaDA exhibits strong scalability, matching the overall performance of ARMs on six tasks.

Table 2. Benchmark Results of Post-trained LLMs. LLaDA only employs an SFT procedure while other models have extra reinforcement learning (RL) alignment. * indicates that LLaDA 8B Instruct, LLaMA2 7B Instruct, and LLaMA3 8B Instruct are evaluated under the same protocol, detailed in Appendix B.5. Results indicated by [†] and [¶] are sourced from Yang et al. (2024) and Bi et al. (2024) respectively. The numbers in parentheses represent the number of shots used for in-context learning. “-” indicates unknown data.

	LLaDA 8B*	LLaMA3 8B*	LLaMA2 7B*	Qwen2 7B [†]	Qwen2.5 7B [†]	Gemma2 9B [†]	Deepseek 7B [¶]
Model	Diffusion	AR	AR	AR	AR	AR	AR
Training tokens	2.3T	15T	2T	7T	18T	8T	2T
Post-training Alignment pairs	SFT	SFT+RL	SFT+RL	SFT+RL	SFT+RL	SFT+RL	SFT+RL
4.5M							
General Tasks							
MMLU	65.5 (5)	68.4 (5)	44.1 (5)	-	-	-	49.4 (0)
MMLU-pro	37.0 (0)	41.9 (0)	4.6 (0)	44.1 (5)	56.3 (5)	52.1 (5)	-
Hellaswag	74.6 (0)	75.5 (0)	51.5 (0)	-	-	-	68.5 (-)
ARC-C	88.5 (0)	82.4 (0)	57.3 (0)	-	-	-	49.4 (-)
Mathematics & Science							
GSM8K	78.6 (4)	78.3 (4)	29.0 (4)	85.7 (0)	91.6 (0)	76.7 (0)	63.0 (0)
Math	26.6 (0)	29.6 (0)	3.8 (0)	52.9 (0)	75.5 (0)	44.3 (0)	15.8 (0)
GPQA	31.8 (5)	31.9 (5)	28.4 (5)	34.3 (0)	36.4 (0)	32.8 (0)	-
Code							
HumanEval	47.6 (0)	59.8 (0)	16.5 (0)	79.9 (0)	84.8 (0)	68.9 (0)	48.2 (-)
MBPP	34.2 (4)	57.6 (4)	20.6 (4)	67.2 (0)	79.2 (0)	74.9 (0)	35.2 (-)

Scalability & Inference

- Demonstrates consistent **performance scaling** with increased compute (up to 10^{23} FLOPs)
- **Lack of KV caching** results in higher inference latency compared to ARMs
- Results support **feasibility** of diffusion-based generation at LLM scale

Scope of Our Work



Thai Text Summarization Task
Focus on summarizing high-context
Thai news text.



SFT from Pretrained Model
Fine-tune *LLaDA* using
ThaiSum dataset.



Context Length
Map reduce vs Refine



Evaluation
LLM-as-a-Judge

Experimental Setup

Models

Diffusion

`GSAI-ML/LLaDA-8B-Instruct`

- Base
- Fine-Tuned (FT)



Hugging Face

Parameters

- Input tokens: 1024
- Output tokens: 256
- Sample steps: 256
- Temperature: 0.7

Dataset

`nakhun/thaisum`

- 358,868 article-summary pairs
- Fine-Tuning Samples: 40,000
- Source: Thai news media



Compute Resources

Cloud platform: Modal

Fine-tuning:

- GPU: A100–80GB

Inference:

- GPU: A100–40GB (x8)

Gemini

 **TYPHOOON**

`gemini-2.0-flash`

`scb10x/scb10x-llama3-1-typhoon2-8b-instruct`

 **PyTorch**



LangChain

Evaluation Method: LLM-as-a-Judge

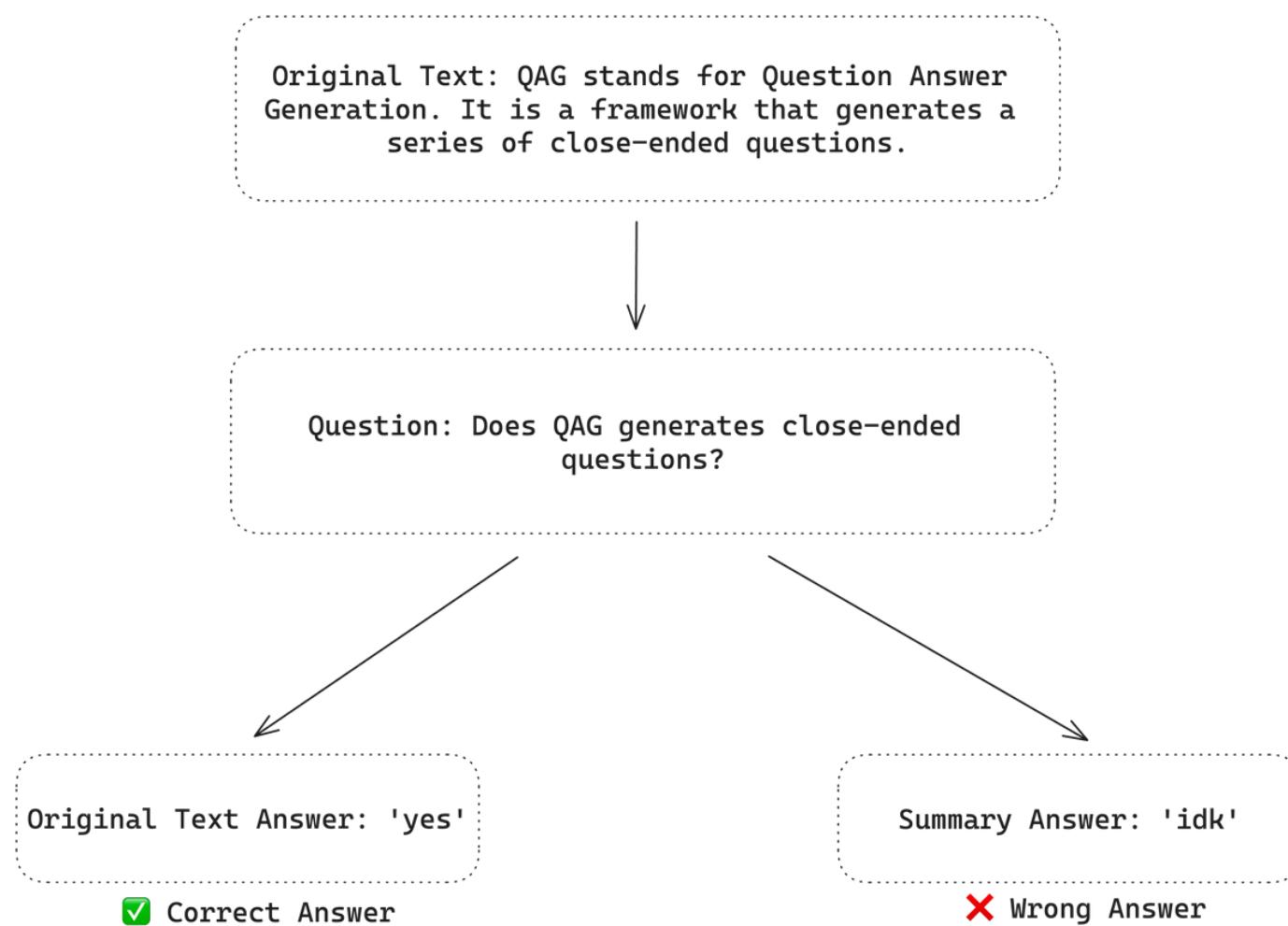
Using QAG (Question-Answer Generation) from DeepEval to evaluate Thai summarization.

100

Averaged over 100 inference runs per model



DeepEval — an open-source LLM evaluation framework developed by Confident-AI



Key Metrics

- **Coverage:** Captures key info from the original text
- **Alignment:** Checks factual consistency with the source

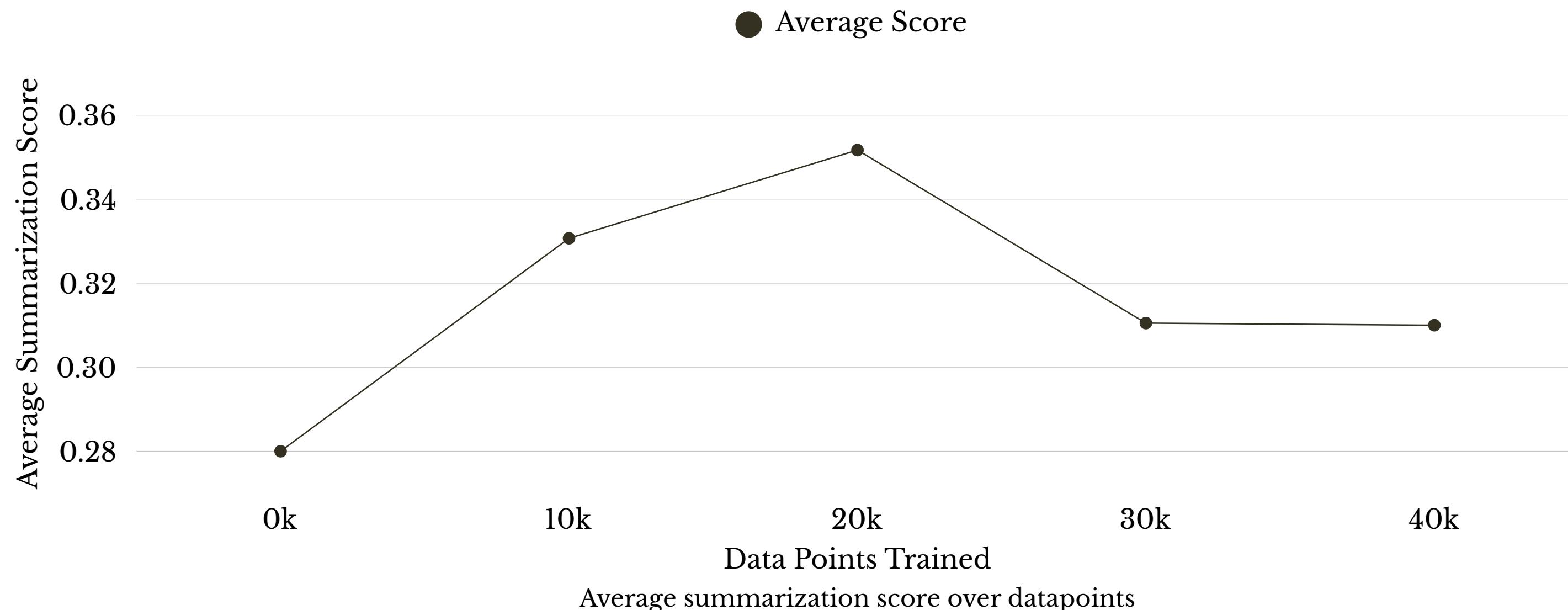
QAG-Based Evaluation Process

1. Generate yes/no questions from both original and summary
2. Answer and compare results: Match / Mismatch
3. Coverage Score: Questions from original text
4. Alignment Score: Questions from summary (detects hallucination)

Final Score: Combine both scores

$$` = \min(\text{Coverage}, \text{Alignment})`$$

Performance Progression



Results: LLaDA, Base vs Finetuned

20,000 Datapoints

	LLaDA-Base	LLaDA-Finetuned
Average Score	0.2828	0.3517

Example of Results

Metrics Summary

- ✓ Summarization (score: 0.7, threshold: 0.5, strict: False, evaluation model: gemini-2.0-flash, reason: The score is 0.70 because the summary omits details about the police chief leading the tourist police team, the tourist police providing services to both Thai and foreign tourists, and the police chief's assurance about the safety of tourism in Chiang Mai. However, the summary does a good job of capturing the core information., error: None)

For test case:

- input: ศูนย์ข่าวภาคเหนือ -12 เม.ย. 2548 ต่าราชการท่องเที่ยวเชียงใหม่พอาสาสมัครชาวต่างชาติออกบริการนักท่องเที่ยว ลงภาพข่าวผู้ก่อการร้ายหลบหนีเข้าเชียงใหม่ พ.ต.ท.ณัฐรุจิ ใจดีภานุวัฒน์ ผู้กำกับการสถานีต่าราชการท่องเที่ยวฯ จ.เชียงใหม่ ได้นำทีมสำรวจท่องเที่ยวและอาสาสมัครชาวต่างประเทศ ออกให้บริการแก่นักท่องเที่ยวชาวไทย และชาวต่างประเทศที่เดินทางมาท่องเที่ยว เชียงใหม่ในปีนี้ รวมทั้งได้ทำความสะอาดในกรุงเทพฯ ปลดล็อกห้องน้ำในชีวิตและทรัพย์สิน โดยได้จัดเจ้าหน้าที่ต่าราชการ ออกปฏิบัติหน้าที่เพื่อป้องกันการก่ออาชญากรรม และการก่อความเดือดร้อน รักษาความปลอดภัย ให้กับนักท่องเที่ยวในย่านประดุจท่าแพ และจุดต่างๆ รอบๆ คุเมืองเชียงใหม่ พ.ต.ท.ณัฐรุจิ กล่าวว่า กรณีที่มีหลายประเทศได้ปล่อยข่าวออกมาว่า เชียงใหม่ไม่มีความปลอดภัย มีผู้ก่อการร้ายหลบเข้ามาอยู่ในเขตพื้นที่เชียงใหม่ ซึ่งเมื่อได้ทำการตรวจสอบแล้ว พบว่าข่าวดังกล่าวไม่เป็นความจริง ดังนั้น จึงขอให้ทุกคนมั่นใจได้ว่า การท่องเที่ยวในเชียงใหม่ มีความปลอดภัย ไม่เกิดเหตุร้ายอย่างแน่นอน

- actual output: ต่าราชการท่องเที่ยวเชียงใหม่ร่วมกับอาสาสมัครต่างชาติ ดูแลนักท่องเที่ยวช่วงสงกรานต์ สร้างความมั่นใจด้านความปลอดภัย หลังมีข่าวลือผู้ก่อการร้ายหลบหนีเข้าเมือง ยืนยันข่าวไม่จริง และเพิ่มมาตรการป้องกันอาชญากรรมรอบคุ้มครอง
- expected output: None
- context: None
- retrieval context: None

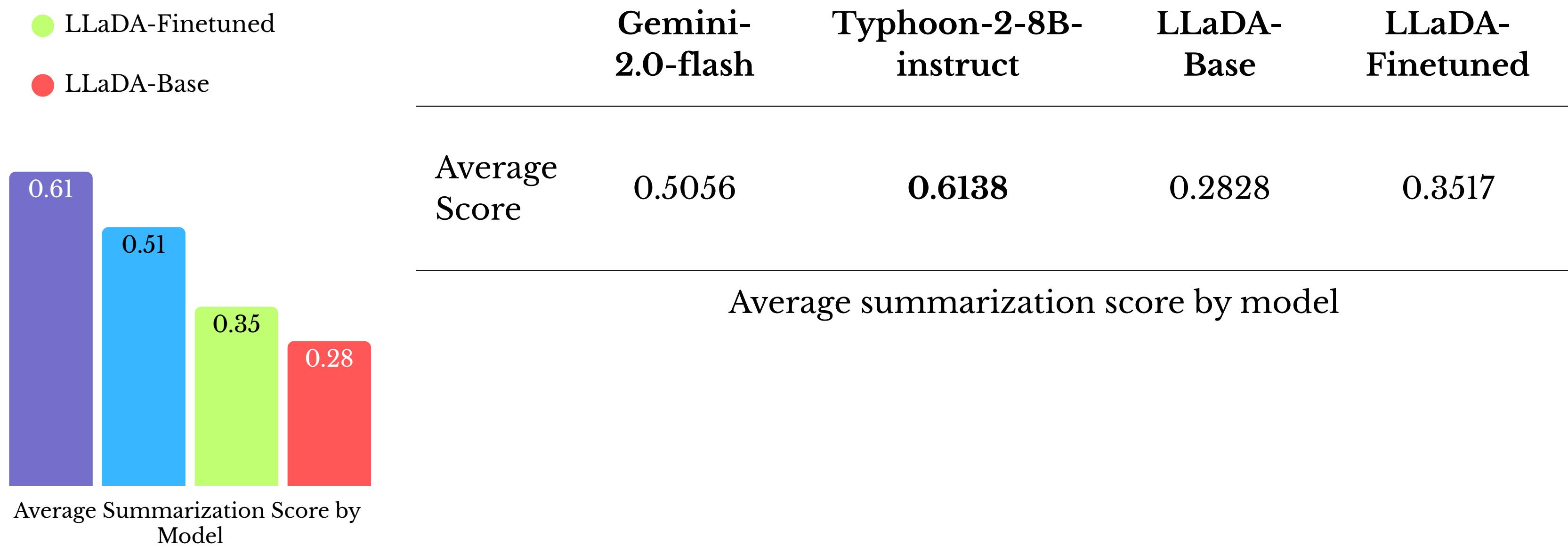
Example of Assessment questions

Assessment Questions

1. Is the Chiang Mai Tourist Police providing services to tourists?
2. Did the Chiang Mai Tourist Police work with foreign volunteers?
3. Did the police chief, Nattawut Chotikarnjanawat, lead the tourist police team?
4. Did the tourist police provide services to both Thai and foreign tourists?
5. Were safety measures implemented for tourists during Songkran in Chiang Mai?
6. Did the police officers perform duties to prevent crime?
7. Did the police address concerns about terrorists in Chiang Mai?
8. Was the news about terrorists in Chiang Mai found to be untrue?
9. Is it safe to travel in Chiang Mai according to the text?
10. Did the Chiang Mai police take measures to ensure the safety of tourists' lives and property?

Results: LLaDA-Base vs LLaDA-Finetuned vs Typhoon vs Gemini

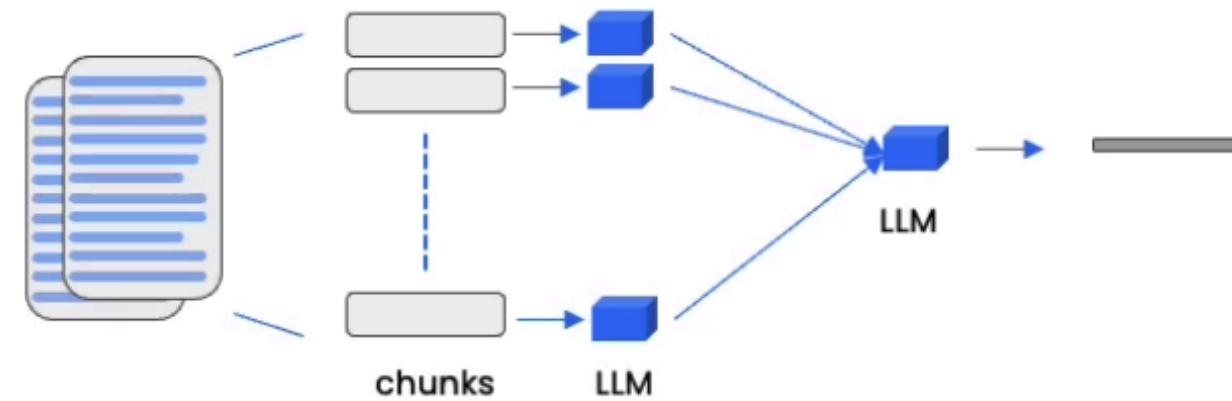
- Typhoon-2-8B-instruct
 - Gemini-2.0-flash
 - LLaDA-Finetuned
 - LLaDA-Base
-



Context-Length Issue

How can LLMs summarize inputs exceeding context limits?

1. Map_reduce



“Parallel summarization, then merging”

Process:

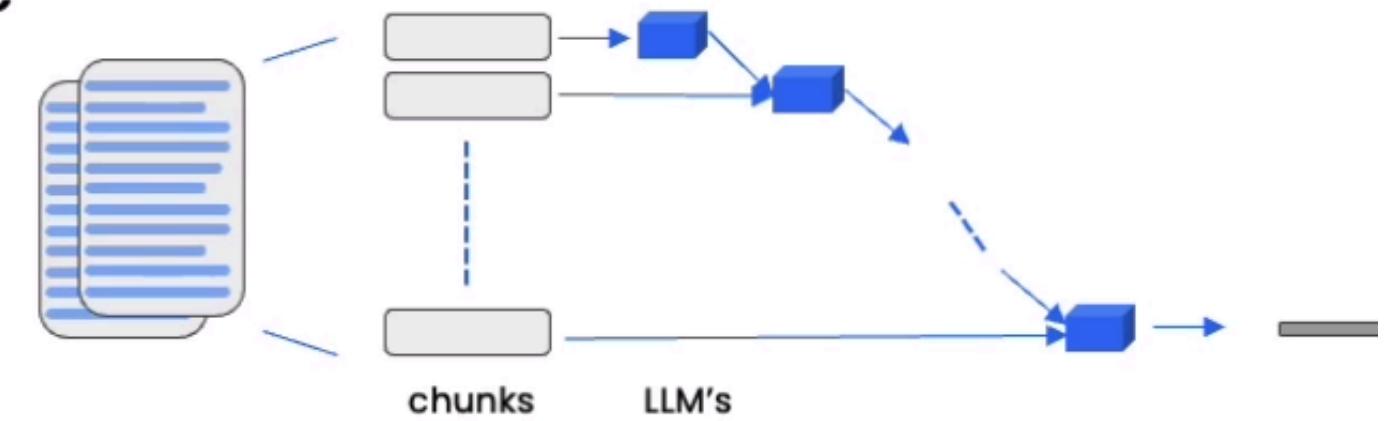
1. **Map:** Summarize each chunk independently
2. **Reduce:** Merge all summaries into one

Pros:

- Parallelizable
- Scales to large corpora

Cons: Loss of global coherence

2. Refine



“Incremental summarization with updates”

Process:

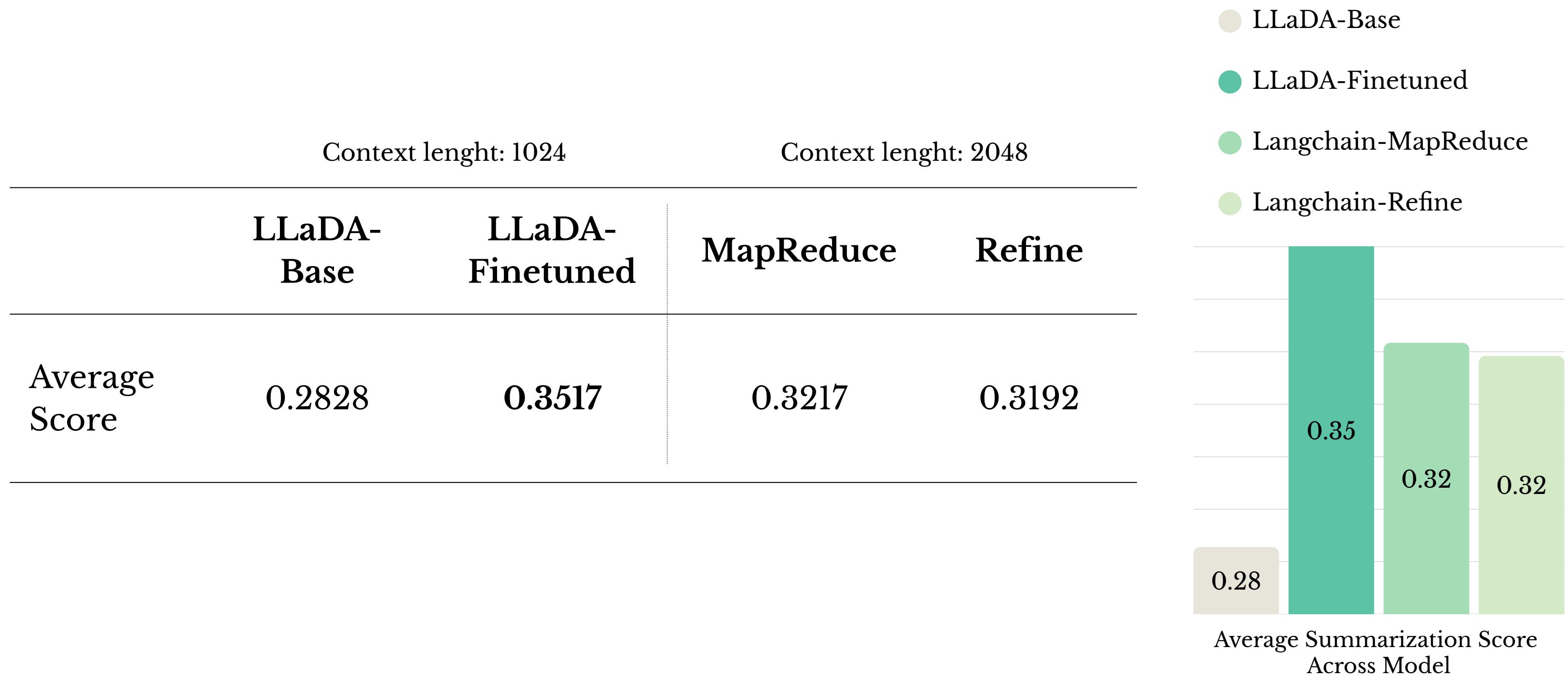
1. **Generate initial summary**
2. **Incrementally update** with each new chunk

Pros:

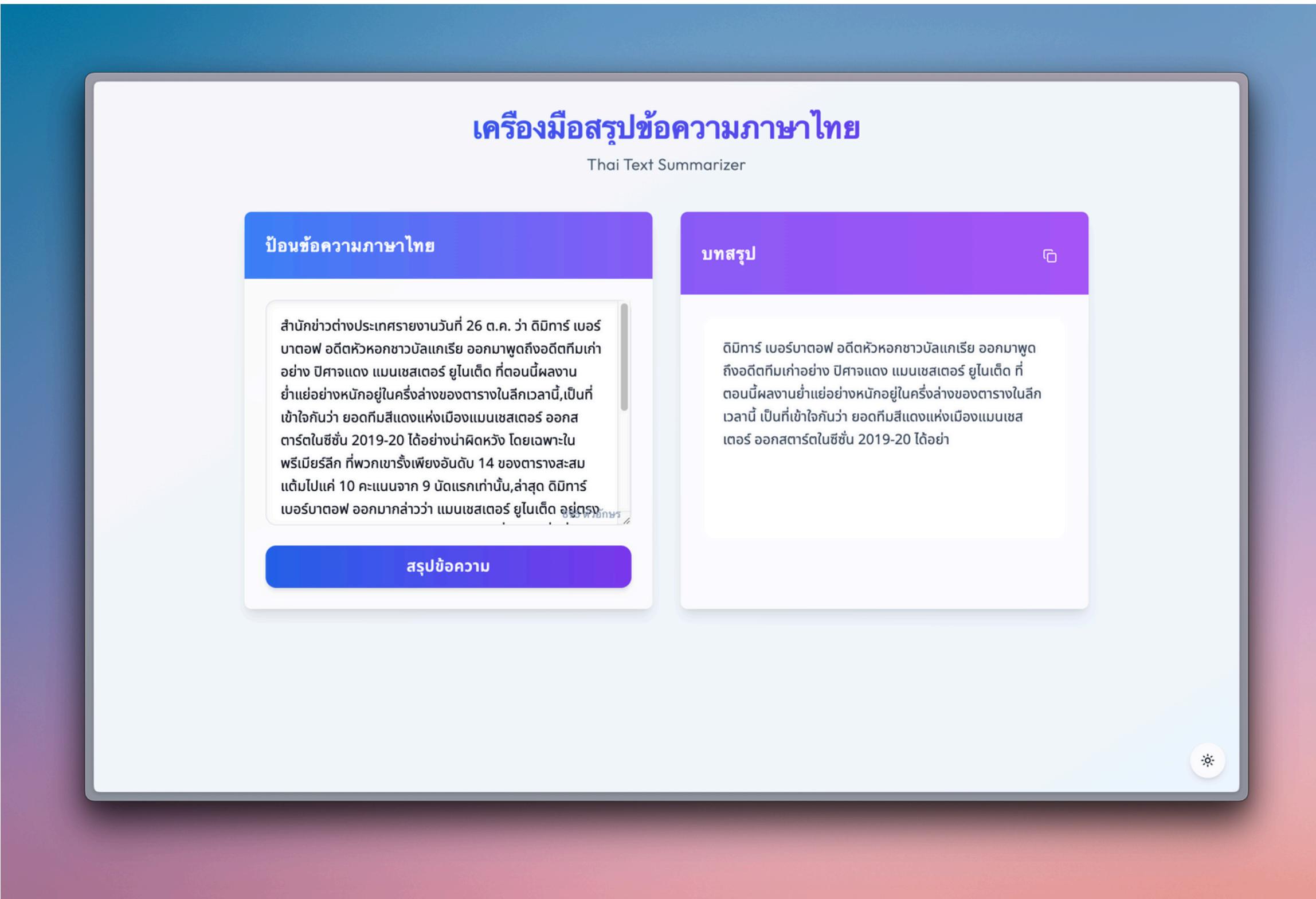
- Maintains global context
- Supports continual refinement

Cons: Sequential → higher latency

Results: MapReduce vs Refine



Application



Execution:

36.33 sec/req

0.72 \$/req

Startup time:

16.26 sec

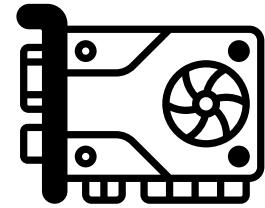


▲Vercel

Conclusion

Model	Mean	Median	SD
LLaDA-Base	0.2828	0.3167	0.2431
LLaDA-Finetuned	0.3517	0.3333	0.2206
Langchain-MapReduce	0.3217	0.3	0.2154
Langchain-Refine	0.3192	0.3167	0.2294
Gemini-2.0-flash	0.5056	0.6	0.2893
Typhoon-2-8B-instruct	0.6138	0.6833	0.3194

Limitation



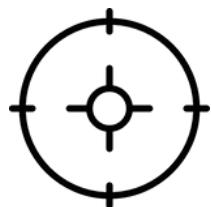
High Resource Requirements

Fine-tuning required A100–80GB; inference on A100–40GB remains constrained.
Experiment was conducted on a small-scale sample, limiting generalizability.



Inference Latency

Multi-step sampling in diffusion models is slower than ARMs due to lack of KV caching.



Evaluation Scope

Focused on LLM-as-a-Judge (QAG-based); other metrics (e.g., ROUGE, BLEU, human eval) were not used.



Context length

Model context window is limited; use of LangChain chunking (e.g., MapReduce/Refine) mitigates this but may hide performance issues on long, unsegmented inputs.

Development Cost

Task	Tool	Cost
Fine-tuning	Nvidia A100-80GB	\$26.81
Inference	Nvidia A100-40GB (x8)	\$41.01
Evaluation	Gemini-2.0-flash	\$1.52
Baseline	Typhoon-8B-Instruct	\$0.01
Baseline	Gemini-2.0-flash	\$0.10
	Total (USD)	\$69.45
	Total (THB)	฿2,284.91

#Gemini API: 16,075 requests

Q & A

Appendices

Inference Cost per Request

Nvidia A100, 40 GB
\$0.000583 / sec

CPU
Physical core (2 vCPU equivalent)
\$0.0000131 / core / sec

Memory
\$0.00000222 / GiB / sec

Compute costs	
	Per hour Per second
GPU Tasks	
Nvidia H100	\$0.001097 / sec
Nvidia A100, 80 GB	\$0.000694 / sec
Nvidia A100, 40 GB	\$0.000583 / sec
Nvidia L40S	\$0.000542 / sec
Nvidia A10G	\$0.000306 / sec
Nvidia L4	\$0.000222 / sec
Nvidia T4	\$0.000164 / sec
CPU	
Physical core (2 vCPU equivalent)	\$0.0000131 / core / sec
<small>*minimum of 0.125 cores per container</small>	
Memory	
	\$0.00000222 / GiB / sec

Total Cost per Request

Execution time: 36.33 sec, Startup time: 16.26 sec

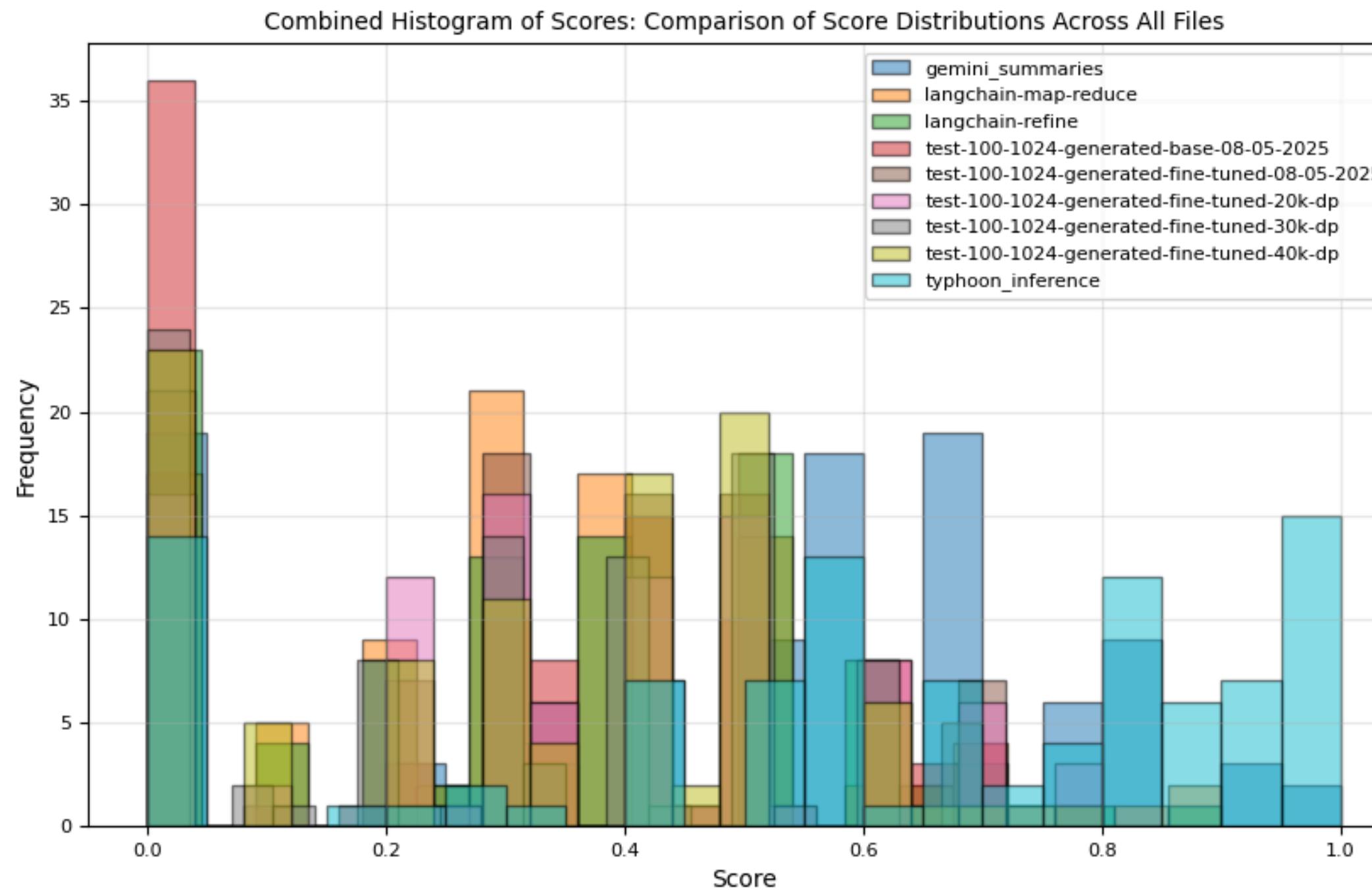
$$= 36.33 * 0.000583 + (16.26 + 36.33) * (0.0000131 + 0.00000222)$$

$$=\$0.0219860688$$

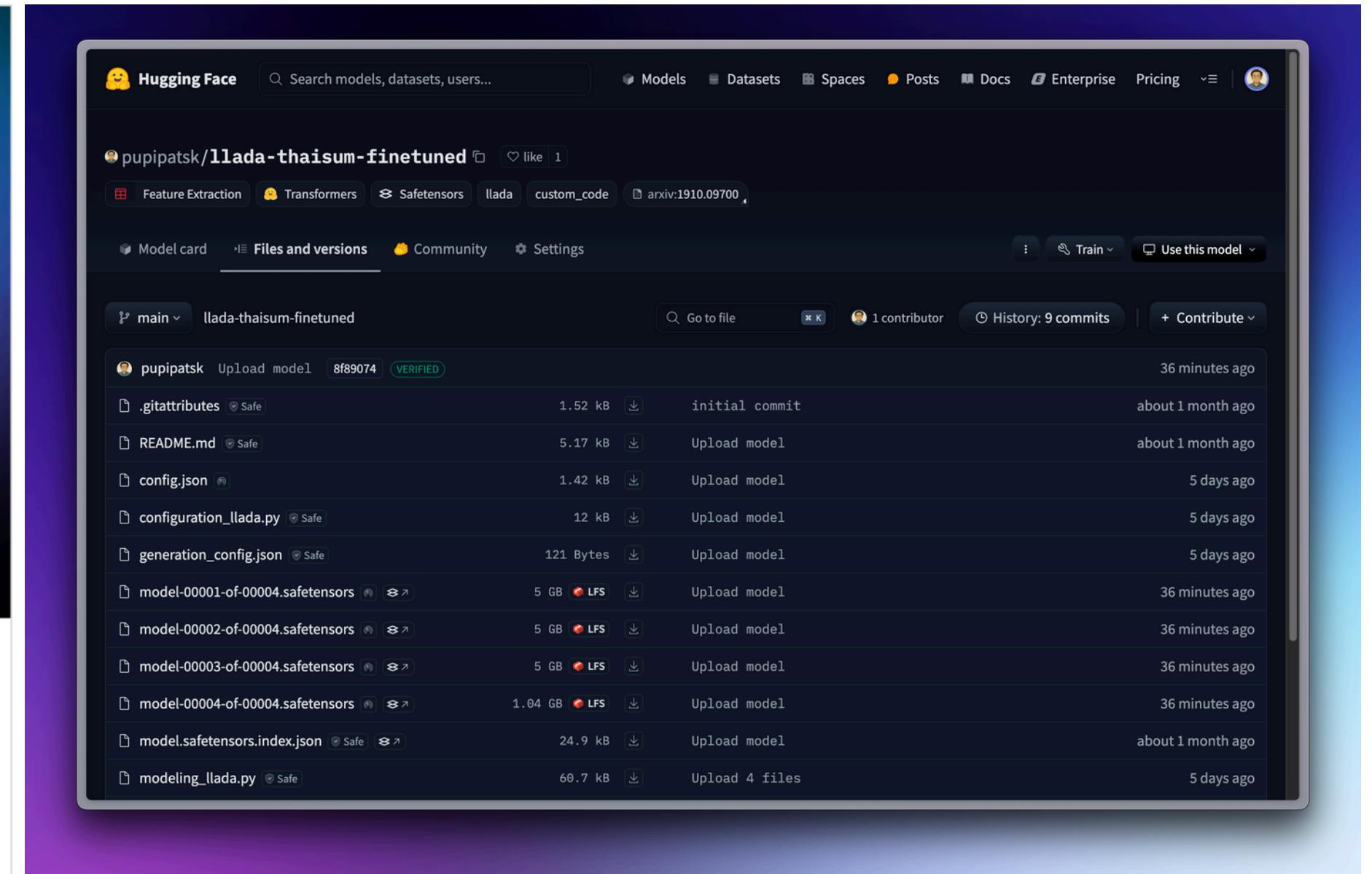
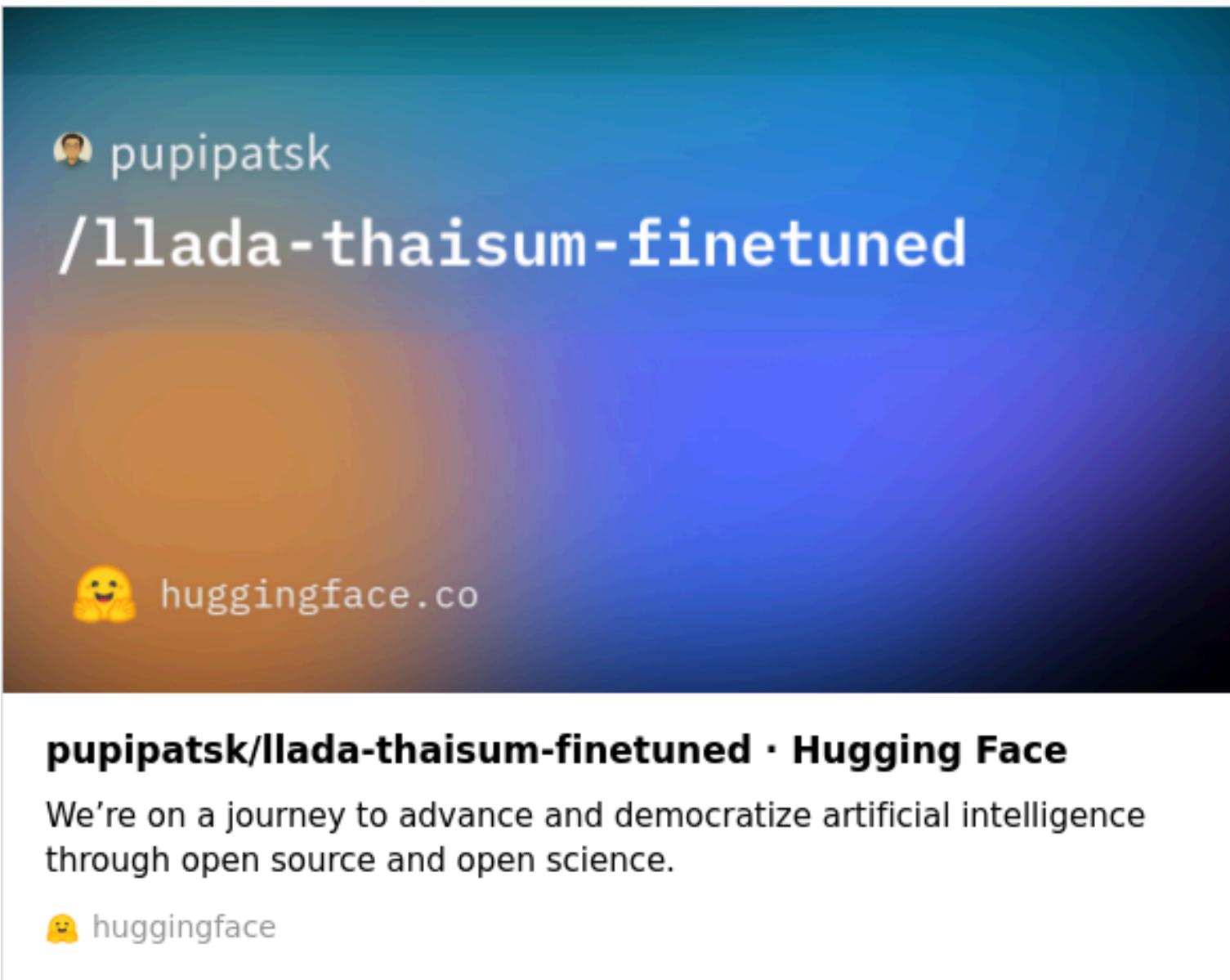
(exchange rate: 32.95 ₩/\$)

= ₩0.724440967 per request

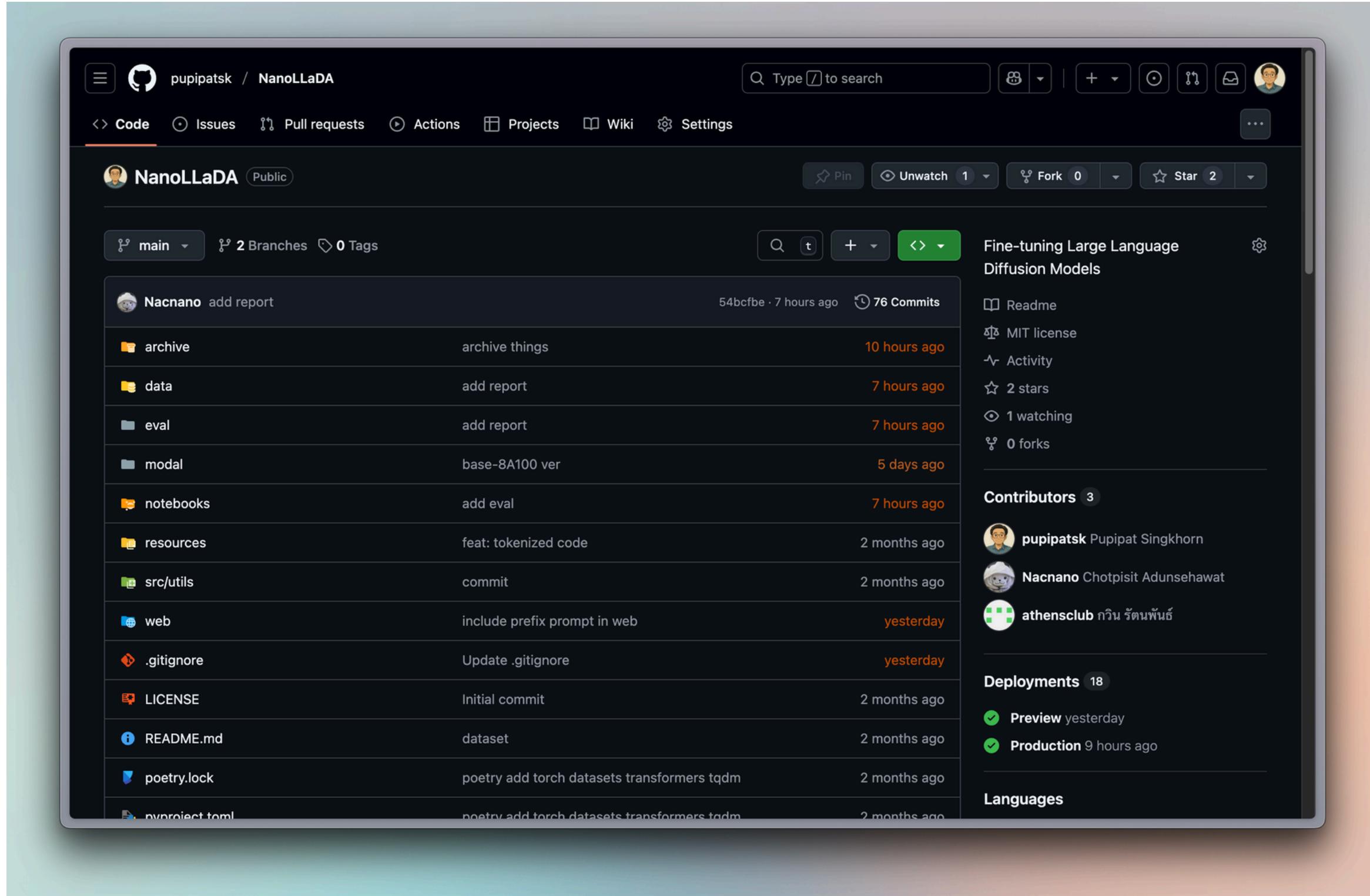
Evaluation Score Histogram



Huggingface



GitHub



Attempt on Smaller Model

```
model_name = "google/mt5-small"
tokenizer = MT5Tokenizer.from_pretrained(model_name)
model = MT5ForConditionalGeneration.from_pretrained(model_name).to(device)
```

Model attempts

- mt5-small
 - mt5-base
 - mt5-large
 - xglm-564M
 - qwen-0.5b