Composing Within Constraints: A Benchmark for Thai Classical Poetry in Large Language Models

Nutchanon Yongsatianchot[1] **Phuvis Kerdpramote**[2]
[1]Faculty of Medicine Siriraj Hospital, Mahidol University, Bangkok, Thailand
[2]Department of Computer Science, School of Science,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
nutchanon.yon@mahidol.ac.th, 66050977@kmitl.ac.th

Abstract

Thai is spoken by tens of millions, yet its literary tradition is largely absent from LLM evaluation. We present ThaiProsody, a benchmark for composing Thai classical poetry. Composing classical Thai verse means choosing words that jointly satisfy several hard formal constraints (fixed syllable counts, written tone-mark positions, heavy/light syllable patterns, and an interlocking web of rhymes) while staying on a given topic. ThaiProsody crosses four canonical verse forms with 40 topics and grades each poem with an open, deterministic rule checker for tone, weight, and rhyme, a syllable counter validated against hand-segmented gold, and a separate content judge. Across 17 models in 24 configurations, the task is far from solved. The best model passes 94.7% of rule checks, yet one stanza in four still breaks a rule; the best open-weight model writes a fully valid stanza only a quarter of the time, and no Thai-trained model more than once in 160 tries. Rhyme is the most frequently broken constraint, and *khlong* and *chan*, the forms that add tone or weight constraints, are the hardest. Thai-trained 8B models stay close to their general base model, while a general 31B model scores more than three times as high as a Thai 30B model.

1 Introduction

Thai is spoken by over sixty million people, yet remains comparatively low-resourced in NLP (Lowphansirikul et al., 2021; Susanto et al., 2025). Its literary tradition is studied even less. Classical Thai poetry sits at the centre of that tradition: a set of strict, centuries-old verse forms. These forms are almost absent from work on large language models (LLMs).

Poetry is a demanding test for an LLM. Recalling or recognizing a poem is one thing; composing one is another. To compose classical Thai verse, a model must choose words that satisfy several hard formal constraints at once. The words must also express a given topic and read as fluent verse. Knowing the tradition is necessary, but not sufficient.

The constraints of Thai classical verse are concrete. Each form fixes a syllable count per line. Some fix where tone marks must fall. One fixes a pattern of light and heavy syllables. All require an interlocking web of rhymes. The constraints interact: a word chosen to complete a rhyme may break a tone or weight pattern. Composing a valid stanza on topic is therefore a search problem under interacting constraints, not a fact to be recalled.

Rule-based and model-based poetry evaluation is mature in other languages, especially Chinese, where prosodic rule-checking is paired with model-based quality scoring (Zhao et al., 2026; Qu et al., 2026). The main Southeast Asian evaluation suite has no poetry task (Susanto et al., 2025), and no benchmark tests Thai verse composition. We close this gap. Our contributions are:

1. **ThaiProsody**, a benchmark for LLM *composition* of Thai classical poetry: four canonical verse forms crossed with 40 topics.
2. An **open meter grader**: deterministic checks of tone-mark placement, light/heavy weight, and rhyme against royal-endorsed prosodic specifications, plus a syllable counter validated against hand-counted gold; the rhyme check and the content judge are also validated against expert labels.
3. A **baseline of 17 models in 24 configurations**, with item-level confidence intervals, a stanza-level validity metric, robustness checks, and an error analysis that localizes difficulty to individual forms, dimensions, and rule positions.
4. **Findings**: the task is far from solved; rhyme is the most frequent failure; *khlong* and *chan* are the hardest forms; Thai specialization does not substitute for scale; and handing over the rules lifts reasoning models close to ceiling but barely

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 197–210
November 6-10, 2026 ©2026 Association for Computational Linguistics

Form	Thai	Ln.	Syll.	Tone	Weight
Khlong	โคลงสี่สุภาพ	8(+2)	30	✓	
Klon	กลอนแปด	4	7–9		
Kap	กาพย์ยานี ๑๑	4	5,6,5,6		
Chan	อินทรวิเชียรฉันท์ ๑๑	4	5,6,5,6		✓

Table 1: The four verse forms. All bind *syllable count* and *rhyme*; *Khlong* adds fixed tone-mark positions, *Chan* adds a light/heavy (ครุ/ลหุ) weight pattern. "8(+2)" = eight mandatory lines plus two optional *khamsoi* (คำสร้อย). Full specs in Appendix A.

moves the others.

2 Thai Classical Prosody

Thai tradition recognizes five classes of verse (ร้อยกรอง): *khlong*, *chan*, *kap*, *klon*, and *rai*. We use the canonical school form of each of the first four and give each a romanized name (Table 1). We leave out *rai*, a non-stanzaic chain of unbounded length with no fixed positional template to verify stanza by stanza. Each chosen form binds a different subset of four constraint dimensions, and together they cover all four.

Four dimensions. **(1) Syllable count.** Every line (*wak*, วรรค) has a prescribed length. Some forms allow a small tolerance (*Klon* permits 7–9 syllables); others are exact. **(2) Tone marks.** *Khlong* fixes the positions of seven *mai-ek* (ไม้เอก) and four *mai-tho* (ไม้โท) marks. Thai writes these marks *explicitly* on the syllable, so the constraint is visible in the surface string. **(3) Light/heavy weight.** *Chan*, adapted from Pali meters of Indic origin (Hudak, 1990), fixes a pattern of heavy (ครุ) and light (ลหุ) syllables. **(4) Rhyme.** A web of obligatory rhymes (*sampat*, สัมผัส) links specific syllables: the last syllable of one line must rhyme with a designated interior syllable of another, chaining the lines of a stanza together.

Density and verifiability. Two properties make these forms a clean testbed. First, the constraints are *bundled*: one compact *Khlong* stanza fixes 30 syllable slots, 11 tone-mark positions, and three rhyme links at once. Second, because tone is written and syllabification is well defined, three of the four dimensions (tone, weight, rhyme) are checkable from the text without a pronunciation lexicon. This enables deterministic, reproducible grading (Section 4.1).

3 Related Work

Computational poetry evaluation. Chinese classical poetry offers the closest precedents. CCiV (Zhao et al., 2026) grades LLM-generated *Ci* poetry by pairing rule-based structural and tonal checking with a model-based quality judge, the two-layer design we adopt, and finds tonal adherence harder than structural compliance. PoeTone (Qu et al., 2026) studies constrained generation of *Songci*, and SongNet (Li et al., 2020) reports automatic format and rhyme F1 for rigidly formatted verse. CCPM (Li et al., 2021) tests classical-poem understanding as four-way multiple choice. For English, Deepspeare (Lau et al., 2018) jointly models meter and rhyme and finds that near-human meter and rhyme do not make generated sonnets match human ones in readability and emotion; Walsh et al. (2024) benchmark poetic-form recognition, and POEMetric scores LLM composition in English fixed forms partly with an LLM judge (Li et al., 2026), whereas we verify meter deterministically.

Metrical poetry beyond Chinese and English. Sanskrit meters, the ancestors of Thai *chan*, are scanned by rule-based tools that classify each syllable as heavy or light and match the pattern to a known meter (Terdalkar and Bhattacharya, 2023; Neill, 2023). With LLMs, only 31% of GPT-4o's one-shot Sanskrit verses satisfy the Anushtubh meter without constrained decoding (Jagadeeshan et al., 2026), and Kranti and Vajjala (2026) use Telugu metrical poetry to probe LLM rule following. Fann or Flop benchmarks Arabic poetry understanding, including prosody (Al Ghallabi et al., 2025). Constrained generation more broadly ranges from hard decoding constraints for meter and rhyme (Ghazvininejad et al., 2016) to compositional constraint benchmarks (Yao et al., 2024).

LLM-as-judge. Model-based judges agree well with humans on many tasks (Zheng et al., 2023; Kim et al., 2024), but carry biases. For poetry, Ma et al. (2026) report an "echo-chamber" effect: LLM judges reward outputs that match statistical patterns yet break strict prosodic rules. This motivates keeping a deterministic meter score as the primary signal and validating any judge against human labels.

Thai NLP. Thai LLM evaluation has grown (Susanto et al., 2025; Kim et al., 2025), but covers knowledge, reasoning, and cultural alignment, not poetry. Concurrent work tests LLM comprehension

of five classical Thai literary works through question answering (Triyason, 2026), but no benchmark tests whether models can compose Thai verse that obeys its meter. Rule-based Thai prosody checking *does* exist in the khavee module of the PyThaiNLP toolkit (Phatthiyaphaibun et al., 2023),[1] which we build on for tone and weight verification.

4 Benchmark Construction

Task. The prompt gives a form, a topic, and a stanza count, and the model writes a new poem. Prompts are zero-shot and minimal (no examples, no rule explanations, no syllable counts), so we test whether the model *knows* a form, not whether it can follow a handed-over specification; Section 6 reports the contrast. The system prompt asks for verse only: "คุณเป็นกวีผู้เชี่ยวชาญฉันทลักษณ์ไทย ตอบเป็นบทร้อยกรองเท่านั้น ไม่ต้องอธิบาย" ("You are an expert in Thai prosody; answer in verse only, no explanation").

Data. The composition set crosses the four forms with 40 topics, 160 items in all. The topics are short, hand-picked themes suited to classical verse, drawn from six categories: nature (13), abstract ideas (11), emotion (6), everyday life (5), seasons (3), and festivals (2). Every form is prompted on every topic, so form comparisons are topic-matched, and topics carry no keyword constraints: the topic varies the content while the form fixes the constraint structure. Each stanza yields dozens of rule decisions, so 40 stanzas per form give several hundred decisions per form.

4.1 Grading

Table 2 lists the three instruments. Only the rule checker and the syllable counter enter the meter score; the content judge is reported separately.

Meter score. We grade four dimensions (syllable count, tone marks (เอก/โท), light/heavy weight (ครุ/ลหุ), and rhyme (สัมผัส)), each as a pass rate in $[0, 1]$, the fraction of that dimension's rules satisfied. **Tone** and **weight** use khavee's verifier, with a dead-syllable fallback for tone and two weight overrides the library gets wrong (an open syllable in long -อ is heavy; the negation บ is light); **rhyme** uses khavee's rime check plus fallbacks for leading-consonant (อักษรนำ) and silent-final (การันต์) words and for open ไ-/ใ- syllables, which the library misreads. These checks involve no model.

[1] https://pythainlp.org/docs/5.0/api/khavee.html

Instrument	Method	Validation
Rule checker (tone, weight, rhyme)	Deterministic: khavee rules plus rime fallbacks	Tone 11/11 and weight 46/46 checks correct on hand-checked canonical verse; rhyme P 1.00, R 0.94 on 44 expert-labeled pairs
Syllable counter (count)	Claude Opus 5.5, cached count-of-record	Exact on 212 hand-counted test items; swapping in a dictionary tokenizer barely changes the ranking
Content judge (not in meter)	Claude Opus 5.5, binary topicality and coherence	Cohen's $\kappa = 0.64$ / 0.56 (on-topic / not word salad) against two raters' adjudicated labels on 132 poems (raters with each other: 0.49 / 0.44; GPT-5.5: 0.64 / 0.00)

Table 2: Grading instruments. Details in Appendices C and D.

Count is the one dimension no rule-based Thai syllabifier handles reliably on classical verse: a dictionary tokenizer agrees with validated counts on only 84% of lines, under-counting implicit-vowel and Sanskrit–Pali syllables. We therefore take the count from a Claude Opus 5.5 syllable counter with a fixed prompt, whose cached output is the *count-of-record* used in scoring (Appendix C). It counts all 212 of our hand-counted test items correctly, and every count is cached, so scores reproduce without further API calls. It is a model, so we weight count least and check that it does not drive the results: substituting the dictionary tokenizer shifts no configuration by more than 0.021, and the only change in the ranking swaps two configurations at the bottom that are 0.002 apart (Appendix G). The meter score is a weighted mean of the dimensions a form binds, weighting the form-defining constraint most: *Khlong* count/tone/rhyme $= .25/.40/.35$; *Klon* and *Kap* count/rhyme $= .40/.60$; *Chan* count/weight/rhyme $= .20/.50/.30$. Equal weights give nearly the same ranking (Spearman $\rho = 0.99$), swapping only near-tied configurations at the bottom. Before grading we normalize the output (strip reasoning traces, code fences, markdown, and stanza labels). Models lay poems out differently (one *wak* per line, or two), so we grade each output under both segmentations and keep the higher score.

Valid stanzas. The meter score counts individual rule checks. A reader judges a stanza as a whole, and a stanza that breaks any one rule is flawed. We therefore also report the *valid-stanza rate*: the frac-

tion of items whose poem satisfies every bound constraint of its form.

Content judge. A Claude Opus 5.5 judge (prompt in Appendix C) flags whether a poem is on-topic and whether its Thai is coherent rather than degenerate. It is a floor check, not an aesthetic judgment. We chose Claude over GPT-5.5 by measuring both against human labels: GPT-5.5 never assigns the lowest coherence score, so it cannot flag degenerate output (Appendix D).

Scoring and completeness. An output is either *graded* or a *non-delivery* (empty or token-runout), which scores 0: a model that exhausts its generation budget (20–24k tokens, which a reasoning model's chain of thought also consumes) fails the item. A serving failure (timeout, server error, or rejected request) is an infrastructure error, not a model failure; we retry it until the item completes, so every configuration is evaluated on all 160 items. We report means with item-bootstrap 95% confidence intervals (1000 resamples).

5 Experimental Setup

We select models to span four groups, so that a score can be read against three axes: access, scale, and Thai specialization. **(1) Frontier closed-source** models set the ceiling: GPT-5.5 (low/medium/high reasoning), GPT-5.4-mini, Gemini 3.5 Flash, Claude Opus 4.7, and Claude Sonnet 4.6. **(2) Frontier open-weight** models ask whether openly released systems close the gap: DeepSeek V4 (Pro and Flash), GLM 5.1, MiniMax M2.7, and Kimi K2.6. **(3) Smaller open-weight** models probe scale: Gemma 4 31B and Qwen3.6 35B-A3B. **(4) Thai-trained** models ask whether Thai specialization helps on a Thai task: the 8B OpenThaiGPT, Typhoon-S, and THaLLE, all built on ThaiLLM-8B, a Thai continued pretraining of Qwen3-8B-Base, and served through ThaiLLM,[2] plus Typhoon 2.5 30B, built on Qwen3-30B-A3B. As a control we also run their general base, Qwen3-8B (Yang et al., 2025) (lineage in Appendix E). Each item is run once; Appendix G repeats full runs for eight configurations. Counting GPT-5.5's three efforts separately gives 19 native configurations of 17 models; thinking-disabled runs of the five open reasoning models bring the total to 24.

[2] https://thaillm.or.th

6 Results and Analysis

The frontier cannot do it yet. Table 3 reports composition. The best model, GPT-5.5 (high), passes 94.7% of rule checks, and confidence intervals separate the frontier tier cleanly from the rest. More reasoning helps monotonically ($0.800 \rightarrow 0.896 \rightarrow 0.947$ for low/medium/high). At the stanza level the gap is plain: the best model's poem breaks at least one rule for 39 of 160 items; Gemma 4 31B, the best open-weight model, writes a fully valid poem for 25% of items; and no Thai-trained model does so even once. At the bottom, four open reasoning models (Kimi K2.6, MiniMax M2.7, Qwen3.6, GLM 5.1) deliver a poem for only a third to a half of the items: their chains of thought exhaust the generation budget (Appendix F.1). Content is rarely the obstacle: the judge finds at least 95% of every model's delivered poems on-topic and coherent (Qwen3.6: 89%), so meter, not topic or fluency, is what they miss.

Scale, not Thai specialization. The three Thai 8B models score 0.258, 0.246, and 0.155; their general base, Qwen3-8B, scores 0.233. Thai continued pretraining and instruction tuning move two of them by less than 0.03, and THaLLE, which fails to deliver a poem for 18% of items, falls lower. Scale separates models far more: at about the same size, the general Gemma 4 31B scores 0.729 and the Thai Typhoon 2.5 30B 0.219.

Knowing the rules versus satisfying them. Handing over the rules separates knowledge from ability. We appended each form's full positional specification to the system prompt for one model per group (Appendix F.2). The reasoning models gain sharply: GPT-5.5 (medium) rises from 0.896 to 0.984 (valid stanzas 58% to 91%) and DeepSeek V4 Flash from 0.415 to 0.619. The non-reasoning models barely move: Gemma 4 31B 0.729 to 0.749 and OpenThaiGPT 8B 0.246 to 0.256. At the frontier the zero-shot gap is mostly missing knowledge of the form; below it, the models lack the ability to satisfy the constraints jointly, which the rules alone do not supply.

Rhyme fails most; *Khlong* and *Chan* are hardest. Among strong composers (Table 4), rhyme is the most failed dimension in three forms and ties with tone in *Khlong*; the two *Chan* rhyme links fail in 59% and 55% of their stanzas. *Klon* is nearly solved (count 1%, rhyme 13% error). Validity concentrates the difficulty in the two forms that add a second kind

Model	n	dlv.	Meter [95% CI]	Valid	Top.	Coh.
GPT-5.5 (high)	160	1.00	0.947 [.926,.967]	76	100	100
GPT-5.5 (medium)	160	1.00	0.896 [.865,.923]	58	100	100
Gemini 3.5 Flash	160	1.00	0.820 [.789,.851]	38	98	98
GPT-5.5 (low)	160	1.00	0.800 [.762,.836]	38	100	100
Gemma 4 31B	160	1.00	0.729 [.694,.762]	25	100	100
Claude Opus 4.7	160	1.00	0.715 [.678,.748]	24	100	100
GPT-5.4-mini	160	1.00	0.588 [.544,.631]	14	100	100
DeepSeek V4 Pro	160	0.93	0.496 [.453,.538]	9	93	93
Claude Sonnet 4.6	160	1.00	0.425 [.383,.466]	6	100	100
DeepSeek V4 Flash	160	0.93	0.415 [.372,.460]	4	92	92
Typhoon-S 8B TH	160	1.00	0.258 [.231,.282]	0	99	96
OpenThaiGPT 8B TH	160	0.98	0.246 [.219,.272]	0	98	95
Qwen3-8B	160	1.00	0.233 [.205,.260]	1	98	98
Typhoon 2.5 30B TH	160	1.00	0.219 [.192,.246]	0	100	99
THaLLE 8B TH	160	0.82	0.155 [.131,.182]	0	82	82
Kimi K2.6	160	0.54	0.143 [.111,.180]	0	54	54
MiniMax M2.7	160	0.41	0.093 [.071,.116]	0	41	41
Qwen3.6 35B-A3B	160	0.33	0.091 [.067,.116]	0	30	29
GLM 5.1	160	0.35	0.087 [.061,.117]	0	35	35

Table 3: Composition leaderboard: configurations evaluated on all 160 items, sorted by meter score. *Meter* = mean over the 160 items (non-delivery = 0) under the count-of-record, with item-bootstrap 95% CIs; dlv. = delivery rate; *Valid* = % of stanzas that satisfy every constraint of their form (non-delivery counting as invalid). *Top.* and *Coh.* = % of poems judged on-topic and coherent (Section 4.1), a non-delivery counting as neither; the judge is Claude Opus 5.5 (Appendix D); one Qwen3.6 output, a leaked reasoning trace the judge declined, counts as neither. TH = Thai-trained.

Form	count	tone	rhyme	weight
Khlong	18	26	26	—
Klon	1	—	13	—
Kap	9	—	30	—
Chan	18	—	57	23

Table 4: Per-dimension *error rate* (% of rule-checks failed) for strong composers (GPT-5.5×3, Gemini, Opus 4.7, Gemma, GPT-5.4-mini).

of constraint: GPT-5.5 (high) writes a fully valid stanza for 35 and 33 of 40 *Klon* and *Kap* topics, but for only 26 *Khlong* and 27 *Chan* topics, and no model outside GPT-5.5 manages more than seven valid stanzas of either form. Appendix H shows what each failure type looks like, and a supplementary study outside the benchmark (Appendix I) applies the grader to filling in and repairing unpublished stanzas, where the same ranking holds.

Robustness. The ranking does not depend on our design choices (Appendix G): equal dimension weights and the dictionary-tokenizer count reproduce it almost exactly (Spearman $\rho = 0.99$ and 1.00), swapping only near-tied configurations at the bottom, and the valid-stanza rate, zero for the nine weakest configurations, agrees at $\rho = 0.95$. Across three full runs of eight configurations the ranking is identical, and most means move by 0.001–0.014 (standard deviation), well inside the gaps between tiers; the two models served through third-party providers drifted between runs months apart. With 40 topics per form the benchmark separates capability tiers, which differ by tens of points; it does not support fine distinctions between adjacent configurations.

7 Conclusion

We presented ThaiProsody, a benchmark for LLM composition of Thai classical poetry. Composing valid verse is a constraint-satisfaction problem that knowing the tradition does not solve. An open meter grader, deterministic except for a validated syllable counter, lets us score it cheaply and reproducibly. Across 17 models, the task is far from solved: even the best model breaks a rule in one stanza in four and every other model in most, rhyme is the most frequent failure, *Khlong* and *Chan* are the hardest forms, and Thai specialization does not substitute for scale. We release the benchmark, the grader, the validation labels, and all model outputs.[3]

[3] https://github.com/yongsa-nut/ThaiProsody. Data and model outputs are released under CC BY 4.0 and code under the MIT license; the data contain no personal information or offensive content.

Limitations

The content judge is validated against two native speakers' labels on 132 poems. Against their adjudicated labels it reaches binary Cohen's $\kappa = 0.64$ (topicality) and 0.56 (coherence), and the raters agree with each other at only 0.49 and 0.44, so even the binary content labels are only moderately reliable. It is a floor check for off-topic and degenerate output; we do not evaluate aesthetic quality, which calls for expert literary judgment. The syllable counter is a model: though exact against our hand-segmented gold, it can still err on rare forms, so count is the lowest-weighted dimension, and positional tone and weight checks locate their slots with the dictionary syllabifier, which can misplace a slot when it miscounts a line. Rhyme grading covers within-stanza links only. Each main-table configuration is run once per item, so its confidence interval reflects item sampling, not decoding variance; our repeated runs cover a subset of models. The benchmark covers four forms of a much larger tradition, with 40 topics per form. Finally, results describe the model versions and serving endpoints we accessed; hosted models change over time.

References

Wafa Al Ghallabi, Ritesh Thawkar, Sara Ghaboura, Ketan Pravin More, Omkar Thawakar, Hisham Cholakkal, Salman Khan, and Rao Muhammad Anwer. 2025. Fann or flop: A multigenre, multiera benchmark for Arabic poetry understanding in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20224–20244, Suzhou, China. Association for Computational Linguistics.

Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. 2016. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1191, Austin, Texas. Association for Computational Linguistics.

Thomas John Hudak. 1990. *The Indigenization of Pali Meters in Thai Poetry*. Number 87 in Monographs in International Studies, Southeast Asia Series. Ohio University Center for International Studies, Athens, Ohio.

Manoj Balaji Jagadeeshan, Samarth Bhatia, Pretam Ray, Harshul Raj Surana, Akhil Rajeev P, Priya Mishra, Annarao Kulkarni, Ganesh Ramakrishnan, Prathosh Ap, and Pawan Goyal. 2026. Chandomitra: Towards generating structured Sanskrit poetry from natural language inputs. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 518–534, Rabat, Morocco. Association for Computational Linguistics.

KBTG Labs, Anuruth Lertpiya, Danupat Khamnuansin, Kantapong Sucharitpongpan, Pornchanan Balee, Tawunrat Chalothorn, Thadpong Pongthawornkamol, and Monchai Lertsutthiwong. 2026. THaLLE-ThaiLLM: Domain-specialized small LLMs for finance and Thai – technical report. *arXiv preprint arXiv:2601.04597*.

Dahyun Kim, Sukyung Lee, Yungi Kim, Attapol Rutherford, and Chanjun Park. 2025. Representing the under-represented: Cultural and core capability benchmarks for developing Thai large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4114–4129, Abu Dhabi, UAE. Association for Computational Linguistics.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Chalamalasetti Kranti and Sowmya Vajjala. 2026. MetricalARGS: Studying metrical poetry with LLMs. In *Proceedings of the 8th Workshop on Indian Language Data: Resources and Evaluation*, pages 1–13, Palma, Mallorca, Spain. ELRA Language Resources Association (ELRA).

Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1948–1958, Melbourne, Australia. Association for Computational Linguistics.

Bingru Li, Han Wang, and Hazel Wilkinson. 2026. POEMetric: The last stanza of humanity. In *The Fourteenth International Conference on Learning Representations (ICLR)*.

Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. 2020. Rigid formats controlled text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 742–751, Online. Association for Computational Linguistics.

Wenhao Li, Fanchao Qi, Maosong Sun, Xiaoyuan Yi, and Jiarui Zhang. 2021. CCPM: A Chinese classical poetry matching dataset. *arXiv preprint arXiv:2106.01979*.

Lalita Lowphansirikul, Charin Polpanumas, Nawat Jantrakulchai, and Sarana Nutanong. 2021. WangchanBERTa: Pretraining transformer-based Thai language models. *arXiv preprint arXiv:2101.09635*.

Bolei Ma, Yina Yao, and Anna-Carolina Haensch. 2026. Capabilities and evaluation biases of large language models in classical Chinese poetry generation: A case study on Tang poetry. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 16933–16957, San Diego, California, United States. Association for Computational Linguistics.

Tyler Neill. 2023. Skrutable: Another step toward effective Sanskrit meter identification. In *Proceedings of the Computational Sanskrit & Digital Humanities: Selected Papers Presented at the 18th World Sanskrit Conference*, pages 97–112, Canberra, Australia (Online mode). Association for Computational Linguistics.

OpenThaiGPT and ThaiLLM Team. 2025. OpenThaiGPT-ThaiLLM-8B-v7.2-Research-Preview: A Thai knowledge-focused language model. Hugging Face model card.

Wannaphong Phatthiyaphaibun, Korakot Chaovavanich, Charin Polpanumas, Arthit Suriyawongkul, Lalita Lowphansirikul, Pattarawat Chormai, Peerat Limkonchotiwat, Thanathip Suntorntip, and Can Udomcharoenchaikit. 2023. PyThaiNLP: Thai natural language processing in Python. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 25–36, Singapore. Association for Computational Linguistics.

Kunat Pipatanakul, Potsawee Manakul, Natapong Nitarach, Warit Sirichotedumrong, Surapon Nonesung, Teetouch Jaknamon, Parinthapat Pengpun, Pittawat Taveekitworachai, Adisai Na-Thalang, Sittipong Sripaisarnmongkol, Krisanapong Jirayoot, and Kasima Tharnpipitchai. 2024. Typhoon 2: A family of open text and multimodal Thai large language models. *arXiv preprint arXiv:2412.13702*.

Kunat Pipatanakul and Pittawat Taveekitworachai. 2026. Typhoon-S: Minimal open post-training for sovereign large language models. *arXiv preprint arXiv:2601.18129*.

Zhan Qu, Shuzhou Yuan, and Michael Färber. 2026. PoeTone: A framework for constrained generation of structured Chinese Songci with LLMs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32764–32772. Association for the Advancement of Artificial Intelligence. Issue 39.

SCB 10X. 2025. Typhoon2.5-Qwen3-30B-A3B. Hugging Face model card.

Yosephine Susanto, Adithya Venkatadri Hulagadri, Jann Railey Montalan, Jian Gang Ngui, Xianbin Yong, Wei Qi Leong, Hamsawardhini Rengarajan, Peerat Limkonchotiwat, Yifan Mai, and William Chandra Tjhi. 2025. SEA-HELM: Southeast Asian holistic evaluation of language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12308–12336, Vienna, Austria. Association for Computational Linguistics.

Hrishikesh Terdalkar and Arnab Bhattacharya. 2023. Chandojnanam: A Sanskrit meter identification and utilization system. In *Proceedings of the Computational Sanskrit & Digital Humanities: Selected Papers Presented at the 18th World Sanskrit Conference*, pages 113–127, Canberra, Australia (Online mode). Association for Computational Linguistics.

ThaiLLM. 2025. ThaiLLM-8B. Hugging Face model card.

Tuul Triyason. 2026. Lilit to Ramakien: A five-work benchmark for LLM understanding of classical Thai literature. In *Proceedings of the 14th International Conference on Advances in Information Technology (IAIT '26)*, pages 1–9. Association for Computing Machinery.

Melanie Walsh, Anna Preus, and Maria Antoniak. 2024. Sonnet or not, bot? Poetry evaluation for large models and datasets. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15568–15603, Miami, Florida, USA. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shunyu Yao, Howard Chen, Austin W. Hanjie, Runzhe Yang, and Karthik Narasimhan. 2024. COLLIE: Systematic construction of constrained text generation tasks. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Shangqing Zhao, Yupei Ren, Yuhao Zhou, Xiaopeng Bai, and Man Lan. 2026. CCiV: A benchmark for structure, rhythm and quality in LLM-generated Chinese *ci* poetry. *arXiv preprint arXiv:2602.14081*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*, pages 46595–46623.

A Full positional specifications

Khlong (โคลงสี่สุภาพ): 8 mandatory lines (syllables 5,2,5,2,5,2,5,4) plus two optional 2-syllable คำสร้อย; seven เอก and four โท at fixed positions; three rhyme links. *Klon* (กลอนแปด): four lines of 7–9 syllables; rhyme from line-end to syllable 3 or 5 of the next line. *Kap* (กาพย์ยานี ๑๑) and *Chan* (อินทรวิเชียรฉันท์ ๑๑): lines of 5,6,5,6 syllables with two internal rhyme links; *Chan* adds the weight

pattern ค-ค-ล-ค-ค | ล-ล-ค-ล-ค-ค. Specifications were verified against royal-endorsed references (สารานุกรมไทยสำหรับเยาวชนฯ; ราชบัณฑิตยสภา).

B Prompt templates

Compose: "จงแต่ง{form} จำนวน {n} บท ในหัวข้อ "{topic}" ให้ถูกต้องตามฉันทลักษณ์ เขียนแต่ละวรรคขึ้นบรรทัดใหม่". This shares the verse-only system prompt (Section 4). In the rules-in-prompt condition (Appendix F.2), the system prompt additionally carries the form's full specification, rendered from the same specification file the grader uses: syllables per line, bound tone positions, the weight pattern, and the rhyme links.

C Grading prompts

Two grading steps use a language model: the syllable counter (count-of-record, Section 4.1) and the binary content judge. We reproduce both prompts.

Syllable counter. Claude Opus 5.5 (claude-opus-5-5, effort *medium*) is called through the Anthropic API with 50 lines per request and this prompt: "You are an expert in Thai prosody (ฉันทลักษณ์) and phonology. For each numbered วรรค (verse segment) below, count the PRONOUNCED syllables (พยางค์) actually present as written." Conventions given: (1) count by pronunciation, not written characters; (2) a silenced การันต์ final adds no syllable (จันทร์→1, ศักดิ์→1; โทรศัพท์→3); (3) อักษรนำ / อักษรควบไม่แท้ with an inserted half-vowel are *separate* syllables (ขนม→2, สบาย→2, สว่าง→2, ขจัด→2); (4) a true cluster (อักษรควบแท้) is one syllable (ปลา→1, ครวญ→1, พราว→1); (5) a silent leading ห/อ before a low consonant is one syllable (หนา→1, อยู่→1); (6) Sanskrit/Pali by Thai pronunciation (มนุษย์→2, วิทยา→3, ธรรมชาติ→3, นภา→2); (7) count all syllables, including across internal spaces. The output is a JSON array giving, for each line, the dot-separated syllable breakdown and the integer count, which must equal the number of syllables in the breakdown; where they disagree we take the breakdown length. Each distinct line is counted once and cached: 38,749 lines across all composition runs (17,724 of them in the main runs), of which three needed the breakdown rule. On 212 hand-counted test items (106 words, each alone and inside a line: 86 in categories where counting goes wrong, namely silent finals, leading consonants with and without an inserted vowel, true clusters, Pali–Sanskrit forms, and compounds, plus 20 simple controls) the counter is exact on all 212. As a cross-check, Claude Opus 4.7, prompted with the same conventions, independently counted a 12,237-line subset and agrees on 98.8% of it.

Binary content judge. Claude Opus 5.5 (effort *medium*) judges each poem in one API call with this prompt: "You are an expert evaluator of Thai poetry. Rate the poem on two axes, each an integer 1–5. TOPICALITY: does the poem address the given topic? (1 off-topic; 3 passing mention; 5 substantively on-topic). COHERENCE: are the Thai sentences intelligible and connected? (1 word salad / a single word repeated; 3 locally grammatical but disjointed; 5 fully coherent)." Hard constraints: do *not* judge syllable count, tone, ครุ/ลหุ, or rhyme (graded separately); do not penalize rare or archaic vocabulary; output JSON only. For scoring we binarize: topical = topicality ≥ 3, coherent = coherence ≥ 2 (i.e. not word-salad).

D Instrument validation

Human labels. A native Thai speaker with a doctoral degree and the standard Thai secondary-school training in classical prosody labeled 176 items with a written Thai rubric in a self-contained interface: 132 model poems spanning the score range, each rated 1–5 for topicality and coherence with the rubric of the content judge, and 44 word pairs, each judged as rhyming or not by sound. The annotation covers judgments a competent native speaker makes reliably; it does not ask for aesthetic judgment. A second native speaker, the second author, labeled the same 176 items independently with the same rubric and interface. The two raters agree on 42 of the 44 rhyme pairs (Krippendorff's $\alpha = 0.88$). On the poems they use the 1–5 scales differently: the second rater scores lower throughout (median topicality 4 vs. 5, coherence 3 vs. 4), so ordinal agreement on the full scales is near zero ($\alpha = 0.04$ and -0.01). On the binary cuts the benchmark uses they agree on 91% of poems for on-topic and 95% for not word salad (Cohen's $\kappa = 0.49$ and 0.44; both raters label most poems positive, which keeps κ low). Raising the cut to 4 lowers their agreement to $\kappa = 0.13$ and 0.08: the raters disagree mainly about where a well-formed poem sits in the upper half of the scale and agree most on the floor cuts the benchmark uses. Where they fall on opposite sides of a benchmark cut (12 poems for on-topic, 7 for not word salad), the first rater re-examined the

poem with both scores in view and set the label, siding with the second rater in 8 of the 19 decisions. With the poems on which the raters agree, these adjudicated labels cover all 132 poems.

Rule checks. The tone and weight checks were audited on hand-segmented canonical verse before use: the tone check classifies all 11 bound *mai-ek* and *mai-tho* positions of a canonical *Khlong* stanza correctly, and the weight check all 22 pattern positions of a canonical *Chan* stanza plus 24 hand-labeled probe syllables (46/46). The rhyme check is validated against both raters below.

Content judge. We compared three judges under an identical prompt (Table 5): GPT-5.5 (medium reasoning), Claude Opus 4.7, and Claude Opus 5.5 (medium effort), which scores the leaderboard. Against the adjudicated labels, Opus 5.5 reaches $\kappa = 0.64$ on on-topic and 0.56 on not word salad (agreement 95.5% and 97.7%), Opus 4.7 0.78 and 0.66, and GPT-5.5 0.64 and 0.00; each rater's own labels reach 0.69–0.74 and 0.66–0.70 against them. GPT-5.5 never assigns the lowest coherence score: poems the first rater scored 1 receive a mean of 3.5 from GPT-5.5 and 2.0 from Opus 4.7, so GPT-5.5 cannot flag word-salad output at all. Only 5 of the 132 poems are word salad under the adjudicated labels, so that κ moves with a single disagreement. Against the individual raters, Opus 5.5 agrees with the first more than the raters agree with each other and with the second less. All judges disagree with the raters mainly between scores 3, 4, and 5 of well-formed poems, a distinction the benchmark does not use, which is why we binarize.

	Topicality	Coherence	
Comparison	≥ 3	≥ 2	≥ 3
GPT-5.5 vs. adjudicated	0.64	0.00	–
Opus 4.7 vs. adjudicated	0.78	0.66	–
Opus 5.5 vs. adjudicated	0.64	0.56	–
GPT-5.5 vs. rater 1	0.72	0.00	0.24
Opus 4.7 vs. rater 1	0.74	0.74	0.41
Opus 5.5 vs. rater 1	0.72	0.66	0.65
Opus 5.5 vs. rater 2	0.39	0.35	0.11
Rater 1 vs. rater 2	0.49	0.44	0.12

Table 5: Content-judge and inter-rater agreement (Cohen's κ, 132 poems). Topicality ≥ 3 is on-topic, coherence ≥ 2 is not word salad, and coherence ≥ 3 is acceptable. The benchmark uses the first two cuts, for which adjudicated labels exist.

Rhyme checker. On the first rater's 44 labeled pairs, the rhyme checker reaches precision 1.000 and recall 0.941 (TP 32, FP 0, FN 2, TN 10); against the second rater, precision and recall are both 0.969 (one error each way). It never accepts a pair the first rater rejects. Its two misses are true rhymes hidden by silent letters: โทรศัพท์–ทับ (*tho-ra-sap–thap*), where the pair is given as a whole word (the grader checks the final syllable ศัพท์, *sap*, which it accepts), and เกียรติ–เบียด (*kiat–biat*), whose silent final is unmarked. Missed rhymes lower a model's rhyme score, so reported rhyme failure rates are, if anything, slight overestimates.

E Model provenance

Table 6 lists the served checkpoints of the Thai-trained models and their lineage. The three 8B models share the ThaiLLM-8B base, a continued pretraining of Qwen3-8B-Base on about 63B tokens of mostly Thai text (ThaiLLM, 2025). Typhoon-S is post-trained from it directly (Pipatanakul and Taveekitworachai, 2026); OpenThaiGPT (OpenThaiGPT and ThaiLLM Team, 2025) and THaLLE (KBTG Labs et al., 2026) merge it with Qwen3-8B. Typhoon 2.5 is built on Qwen3-30B-A3B (SCB 10X, 2025; Pipatanakul et al., 2024). Qwen3-8B (Yang et al., 2025) is therefore the general model underneath all three 8B models.

Model	Served checkpoint	Lineage
OpenThaiGPT	openthaigpt-thaillm-8b-instruct-v7.2	ThaiLLM-8B merged with Qwen3-8B, then fine-tuned
Typhoon-S	typhoon-s-thaillm-8b-instruct	Post-trained from ThaiLLM-8B
THaLLE	thalle-0.2-thaillm-8b-fa	Equal-weight merge of ThaiLLM-8B, Qwen3-8B, and a finance-tuned Qwen3-8B
Typhoon 2.5	typhoon-v2.5-30b-a3b-instruct	Built on Qwen3-30B-A3B
Qwen3-8B	qwen/qwen3-8b	General base of the three 8B models

Table 6: Thai-trained models and their base. The three 8B models are served through the ThaiLLM gateway, Typhoon 2.5 through its developer's API, and Qwen3-8B through OpenRouter.

	native		thinking off	
Model	dlv.	meter	dlv.	meter
DeepSeek V4 Pro	0.93	0.496	1.00	0.512
DeepSeek V4 Flash	0.93	0.415	1.00	0.472
Qwen3.6 35B-A3B	0.33	0.091	1.00	0.278
GLM 5.1	0.35	0.087	1.00	0.241
Qwen3-8B	1.00	0.233	1.00	0.214

Table 7: Thinking-disabled ablation for the open reasoning models on the 160 composition items (non-delivery $= 0$). *native* columns repeat Table 3; both meters use the count-of-record.

F Ablations

F.1 Thinking disabled

Five open reasoning models (DeepSeek V4 Pro/Flash, GLM 5.1, Qwen3.6, Qwen3-8B) can run with thinking disabled. In native thinking mode, GLM 5.1 and Qwen3.6 usually exhaust the token budget on the chain of thought (delivery 0.35 and 0.33), and the two DeepSeek models occasionally do. With thinking disabled (Table 7) every model delivers all 160 poems, and the meter score rises sharply for GLM 5.1 and Qwen3.6, whose native scores were depressed mainly by non-delivery. For the other three, the change is at most 0.06.

F.2 Rules in the prompt

Table 8 gives the full rules-in-prompt ablation. With the specification in hand, GPT-5.5 satisfies every count and tone rule it is given and nearly every weight and rhyme rule. DeepSeek V4 Flash gains on every dimension but remains far below. Gemma 4 31B and OpenThaiGPT 8B barely change: knowing the rules does not give them the ability to satisfy all of them in one stanza.

G Robustness

Syllable counter. Table 9 recomputes every configuration's meter score with the dictionary tokenizer's syllable count in place of the count-of-record. No score moves by more than 0.021, and the two rankings agree (Spearman $\rho = 0.998$) except that MiniMax M2.7 and Qwen3.6, 0.002 apart at the bottom, change places.

Weights. Re-ranking with equal weights on each form's bound dimensions agrees with the published ranking at Spearman $\rho = 0.993$; the only changes swap configurations at the bottom that differ by at most 0.012 (GLM 5.1, MiniMax M2.7, and Qwen3.6; Kimi K2.6 and THaLLE). Ranking by the valid-stanza rate, which uses no weights, agrees at $\rho = 0.95$; the nine weakest configurations write no valid stanza and tie.

Repeated runs. Table 10 repeats the full composition set three times for eight configurations spanning the leaderboard. The ranking is identical in every run. For six of the eight, the run-to-run standard deviation is a fraction of a single run's confidence interval. DeepSeek V4 Flash and Qwen3-8B, both served by third-party providers through a routing service, scored lower in the two later runs, which agree with each other; the earlier and later runs are months apart, so we attribute the shift to changes on the serving side (for DeepSeek, a lower delivery rate) rather than to decoding variance.

Topic categories. Table 11 breaks the meter score down by topic category. Category means differ by at most 0.02 across all configurations, and within each category the model ranking tracks the overall one; the two festival topics (8 items) are too few to rank the strong composers reliably.

H Qualitative analysis of failures

The error rates in Table 4 count failed rule-checks. This appendix shows what those failures look like, with annotated compositions from three capability tiers: frontier (GPT-5.5, Gemini 3.5 Flash, Claude Opus 4.7), mid (Gemma 4 31B, GPT-5.4-mini, Claude Sonnet 4.6, DeepSeek V4), and the Thai-trained 8B models (OpenThaiGPT, Typhoon-S, THaLLE). For each failure type we show a stanza from the composition set in which that type is the main or only violation. We quote only the relevant lines (*wak*), two per row as in print, with the syllables at issue in bold, then give an English gloss, the violation, and the grader's verdict on the graded stanza (syllable count from the count-of-record of Section 4.1; meter score as in Table 3). Positions are given as line and syllable numbers within the stanza. Where a rate from Table 4 applies, the heading gives it (% of rule-checks failed by the strong composers of that table).

(a) A fully valid stanza. GPT-5.5 (high), *Chan*, topic ป่าใหญ่ยามเช้า (the great forest in the morning).

ป่าใหญ่ก็ไพร**พราย** สิริ**สาย**จะฉาย**ทอง**
แดดเช้าก็เรือง**รอง** ระดะผ่องระยับไพร
"The great forest glimmers; auspicious rays cast their gold; the morning sun glows; bright glints are strewn across the woods."

Model	Meter	Valid	count	tone	weight	rhyme
GPT-5.5 (medium)	.896 → .984	93 → 145	.95 → 1.00	.86 → 1.00	.89 → .97	.84 → .97
Gemma 4 31B	.729 → .749	40 → 52	.86 → .89	.67 → .74	.61 → .72	.63 → .65
DeepSeek V4 Flash	.415 → .619	6 → 36	.62 → .82	.24 → .56	.60 → .73	.27 → .57
OpenThaiGPT 8B TH	.246 → .256	0 → 1	.27 → .23	.31 → .28	.64 → .66	.09 → .14

Table 8: Rules-in-prompt ablation: zero-shot → the same prompt with the form's full specification (syllable counts, tone positions, weight pattern, rhyme links) in the system prompt, 160 items each. *Meter* under the count-of-record; *Valid* = stanzas satisfying every constraint (of 160); per-dimension columns = mean per-poem pass rate over delivered poems (tone: *Khlong* only; weight: *Chan* only).

Model	Record	Tokenizer	Δ
GPT-5.5 (high)	0.947	0.934	−0.014
GPT-5.5 (medium)	0.896	0.885	−0.011
Gemini 3.5 Flash	0.820	0.799	−0.021
GPT-5.5 (low)	0.800	0.790	−0.010
Gemma 4 31B	0.729	0.710	−0.019
Claude Opus 4.7	0.715	0.699	−0.016
GPT-5.4-mini	0.588	0.582	−0.005
DeepSeek V4 Pro	0.496	0.482	−0.013
Claude Sonnet 4.6	0.425	0.429	+0.004
DeepSeek V4 Flash	0.415	0.414	−0.001
Typhoon-S 8B TH	0.258	0.261	+0.003
OpenThaiGPT 8B TH	0.246	0.252	+0.006
Qwen3-8B	0.233	0.234	+0.001
Typhoon 2.5 30B TH	0.219	0.220	+0.001
THaLLE 8B TH	0.155	0.160	+0.004
Kimi K2.6	0.143	0.142	−0.001
MiniMax M2.7	0.093	0.092	−0.001
Qwen3.6 35B-A3B	0.091	0.093	+0.002
GLM 5.1	0.087	0.088	0.000

Table 9: Meter score with the syllable count taken from the count-of-record (Record) versus the dictionary tokenizer (Δ = tokenizer − count-of-record). Max $|\Delta| = 0.021$; Spearman $\rho = 0.998$ between the two rankings.

Model	Run 1	Run 2	Run 3	Mean ± SD	Rank
GPT-5.5 (medium)	0.896	0.888	0.897	0.894 ± 0.004	1/1/1
Gemini 3.5 Flash	0.820	0.818	0.819	0.819 ± 0.001	2/2/2
Gemma 4 31B	0.729	0.755	0.753	0.745 ± 0.014	3/3/3
GPT-5.4-mini	0.588	0.569	0.592	0.583 ± 0.012	4/4/4
DeepSeek V4 Flash	0.415	0.268	0.277	0.320 ± 0.082	5/5/5
Typhoon-S 8B TH	0.258	0.263	0.263	0.261 ± 0.003	6/6/6
OpenThaiGPT 8B TH	0.246	0.247	0.229	0.241 ± 0.010	7/7/7
Qwen3-8B	0.233	0.190	0.201	0.208 ± 0.022	8/8/8

Table 10: Run-to-run replication: three independent generation passes over the 160 composition items with identical decoding settings; run 1 is the original evaluation run and runs 2 and 3 were generated together, months later. Scores are mean meter scores (non-delivery = 0) with the count-of-record in every run. Rank = position among the replicated configurations in runs 1/2/3. The ranking of the 8 configurations is identical in every run (Spearman $\rho = 1.00$). For 6 of the 8 configurations the standard deviation across runs (0.001–0.014) is 3–43% of the item-bootstrap 95% CI half-width of a single run (0.025–0.045). DeepSeek V4 Flash and Qwen3-8B, served by third-party providers through a routing service, score lower in runs 2 and 3 than in run 1, while runs 2 and 3 agree within 0.011; a run-1 versus later-run shift of this size exceeds the single-run CI half-width.

All four lines have the required 5, 6, 5 and 6 syllables, all 22 syllables match the weight pattern H-H-L-H-H | L-L-H-L-H-H (H heavy, L light), and both rhyme links hold (พราย *phrai* with สาย *sai*, ทอง *thong* with รอง *rong*). *Verdict:* count 4/4; weight 22/22; rhyme 2/2; meter 1.00. The light slots hold short syllables such as ก็ *ko*, จะ *cha*, สิ-ริ *si-ri* and ระ-ดะ *ra-da*. In the 27 *Chan* stanzas that GPT-5.5 (high) gets fully right, 121 of the 216 light slots hold one of six short function words (จะ *cha*, ณ *na*, มิ *mi*, ก็ *ko*, ละ *la*, แล *lae*).

(b) One rhyme link short of valid (*Chan* rhyme, 57%). GPT-5.5 (low), *Chan*, topic ฤดูฝน (the rainy season).

ฝนหลั่งละพร่าง**พราย** มิจะ**ขาด**ละสายฝน
ลมเย็นละล่องวน มิจะร้างละกลิ่นดิน
"Rain pours down, glittering, its streams never breaking off; the cool wind drifts and eddies, never without the scent of earth."

The first link requires the last syllable of line 1, พราย (*phrai*), to rhyme with syllable 3 of line 2, which is ขาด (*khat*). A rhyming word, สาย (*sai*), is present but sits in slot 5. *Verdict:* count 4/4; weight 22/22; rhyme 1/2; meter 0.85. Twenty-seven of the stanza's 28 checks pass, yet it is not valid. The pattern recurs among strong composers: of their 77 *Chan* stanzas with perfect count and weight, 22 fail a rhyme link (the two links fail in 59% and 55% of their stanzas). Because validity requires every check to pass, GPT-5.5 (high), with a meter score of 0.947, still writes an invalid stanza for 39 of its 160 items, 13 of them among its 40 *Chan* items.

		strong composers					all
Category	Topics	*Khlong*	*Klon*	*Kap*	*Chan*	all	all
nature	13	.729	.901	.780	.678	.772	.438
abstract	11	.767	.951	.773	.680	.793	.444
emotion	6	.747	.917	.760	.681	.776	.430
everyday life	5	.826	.914	.860	.676	.819	.449
seasons	3	.804	.895	.771	.702	.793	.441
festivals	2	.734	.929	.779	.591	.758	.428
all	40	.760	.920	.784	.676	.785	.440

Table 11: Mean meter score by topic category (each topic is prompted in all four forms). Strong composers = 7 configurations (GPT-5.5×3, Gemini, Opus 4.7, Gemma, GPT-5.4-mini); *all* (last column) = the 19 configurations of Table 3. Non-delivery = 0.

(c) Tone mark one slot early (*Khlong* tone, 26%). Gemini 3.5 Flash, *Khlong*, topic ป่าใหญ่ยามเช้า (the great forest in the morning); the whole stanza.

สุริยงส่องฟ้า เรืองรอง
หมอกป่าพรมยอดทอง เพริศแพร้ว
น้ำค้างหยาดหยดนอง **แหล่งหล้า**
ป่าใหญ่ตื่นนอนแล้ว โลกฟื้นเริงใจ
"The sun lights the sky, radiant; forest mist sprinkles the golden treetops, resplendent; dew drips and pools over the land; the great forest has woken, and the world revives in joy."

In *Khlong*, line 4 (the end of the second row) must end with an *ek* syllable then a *tho* syllable, and line 6 (the end of the third row) with a free syllable then an *ek* syllable (a dead syllable counts as *ek*). The poem ends line 4 correctly (เพริศ *phroet*, a dead syllable, then แพร้ว *phraeo* with *mai-tho*) but gives line 6 the same shape: the *mai-ek* falls on แหล่ง *laeng*, one slot early, and the required *ek* slot holds หล้า *la*, which carries *mai-tho*. *Verdict:* count 8/8; tone 10/11; rhyme 3/3; meter 0.96. This slot (line 6, syllable 2) is the most-failed tone position among strong composers (35%). Gemini 3.5 Flash misses it in 25 of its 40 *Khlong* poems, 20 times with the *mai-ek* on the syllable before.

(d) Near-miss rhyme (*Kap* rhyme, 30%). GPT-5.4-mini, *Kap*, topic ความเศร้าในยามพลัดพราก (sorrow at parting); lines 1–2.

ยามพรากจากสุด**แสน** ใจแห้ง**แล้ง**ไร้แสงหวัง
"Parting brings grief beyond measure; the heart is parched and barren, without a ray of hope."

Syllable 3 of line 2 must rhyme with แสน (*saen*). แล้ง (*laeng*) has the same vowel แ- but closes with ง (*-ng*) instead of น (*-n*), so the rime differs. *Verdict:* count 4/4; rhyme 1/2; meter 0.70.

(e) Hidden syllables in a Sanskrit compound (*Kap* count, 9%). DeepSeek V4 Pro, *Kap*, topic มิตรภาพที่ยั่งยืน (lasting friendship).

มิตรภาพแนบแน่นนาน เปรียบดั่งงานหนักเบาบาง
ร่วมทุกข์ร่วมสุขพลาง มีมิตรพลางสร้างทางใจ
"Friendship, close and lasting, is like a heavy task made light; sharing sorrow and joy, having friends builds the heart's path."

On its own มิตร is one syllable (*mit*), as in line 4, but the compound มิตรภาพ is read *mit-tra-phap*, with an unwritten linking syllable, so line 1 has 6 syllables where *Kap* needs 5. *Verdict:* count 3/4; rhyme 2/2; meter 0.90 (1.00 under the dictionary tokenizer, which splits the word as มิตร-ภาพ, *mit-phap*). Implicit vowels (สว่าง read *sa-wang*) and Sanskrit–Pali words (ปฐพี read *pa-tha-phi*, นัยนา read *nai-ya-na*) cause the same drift elsewhere. About a fifth of strong composers' count failures in *Khlong* (18%) and *Kap* (23%) are lines that the dictionary tokenizer accepts but that are too long as pronounced.

(f) Heavy syllables in light slots (*Chan* weight, 23%). DeepSeek V4 Flash, *Chan*, topic ความหวังในวันใหม่ (hope for a new day).

เมื่อฟ้า**สาง**ส่องแสง ประ**หนึ่ง**แจ้ง**ความ**หวังใหม่
ในวัน**นี้**มีใจ จะ**ก้าว**ไป**ไม่**หวั่นเกรง
"As the sky brightens and the light shines, as if announcing a new hope; on this day, with a willing heart, I will step forward without fear."

Counts and rhymes are all correct, but six of the stanza's eight light slots hold heavy syllables: a final consonant (สาง *sang*, หนึ่ง *nueng*, ความ *khwam*), a long vowel (นี้ *ni*, ก้าว *kao*), or the vowel ไ- (ไม่ *mai*). *Verdict:* count 4/4; weight 16/22; rhyme 2/2; meter 0.86. This is the dominant kind of weight error: 64% of strong composers' weight failures put a heavy syllable in a light slot rather than the reverse.

(g) Degenerate repetition. Typhoon-S 8B, *Kap*, topic พระจันทร์เต็มดวง (the full moon); lines 1–2.

พระจันทร์เต็มดวงลอยเด่นกลางฟ้า
แสง**ส่องส่องส่องส่องส่องส่องส่อง**
"The full moon floats bright in mid-sky; light shines shines shines shines shines shines shines."

Line 2 repeats one syllable seven times. All four graded lines have eight syllables, the length of a *Klon* line, where *Kap* needs 5 and 6, and neither rhyme link holds. *Verdict:* count 0/4; rhyme 0/2; meter 0.00. Repetition of this kind (a syllable four or more times in a row, or a line three or more times) appears in 29% of OpenThaiGPT's delivered poems, 16% of Typhoon-S's and 9% of THaLLE's, and in none from the frontier and mid tiers. On the same item, OpenThaiGPT repeats a single line 190

times until generation is cut off. The line lengths are also typical: 66% of the Thai 8B models' graded *Kap* and *Chan* lines have the 7–9 syllables of a *Klon* line (1% for frontier models).

I Supplementary tasks: filling and repairing stanzas

The benchmark and its leaderboard cover composition. As a supplementary study, we apply the same grader to two tasks that start from a valid stanza. Their items derive from only 51 source stanzas, so we report them separately rather than on the leaderboard. **Cloze** (filling) blanks one or two meter-bearing positions (_____): a rhyme source, a *mai-ek* slot (*Khlong*), or a light or heavy slot (*Chan*). The model returns the completed stanza ("จงเติมคำที่หายไป (_____) ในบทร้อยกรองต่อไปนี้ ให้ถูกต้องตามฉันทลักษณ์และความหมาย แล้วเขียนบทที่สมบูรณ์ แต่ละวรรคขึ้นบรรทัดใหม่"; "Fill in the missing words (_____) in the following verse so that it is correct in prosody and meaning, then write out the complete stanza with each line on a new line"). **Editing** (repairing) injects one violation: a required tone mark is removed (*Khlong*), or the heavy syllable มา (*ma*) replaces a light one (*Chan*). The model returns the corrected stanza ("บทร้อยกรองต่อไปนี้มีข้อผิดพลาดด้านฉันทลักษณ์ จงแก้ไขให้ถูกต้อง แล้วเขียนบทที่แก้แล้วใหม่ทั้งบท แต่ละวรรคขึ้นบรรทัดใหม่"; "The following verse contains a prosodic error. Correct it, then rewrite the whole corrected stanza with each line on a new line"). *Klon* and *Kap* bind no tone or weight rule, so editing covers *Khlong* and *Chan*. Both tasks use the verse-only system prompt of Section 4, and each model keeps its composition settings.

Examples. Cloze (*Klon*): เมื่อลมเช้าเป่ามาเหนือทุ่ง กว้าง / เหมือน เปิด ทาง วัน ใหม่ ให้สด_____ / หอบ กลิ่น ดิน กลิ่น หญ้า มา แนบใจ / ปลุก ฝัน ให้ตื่นรับกับตะวัน ("The morning wind blows over the wide field, / as if opening the way for a new day to be _____, / carrying the scent of earth and grass to the heart, / waking dreams to greet the sun"). The blank ends the second line, so it must rhyme with ใจ (*chai*) at the end of the third; the original has ใส (*sai*, "bright"), which shares its rime *-ai*. Editing (*Khlong*, last two lines): the input reads ประโยชน์จึงงอกได้ / ทัวทั้งหมู่ชน ("so benefit can grow / for all the people"), where ทั่ว (*thua*, "all, throughout") has lost its *mai-ek*; the first syllable of the last line requires one, and the minimal fix restores ทั่ว.

Source stanzas. To limit memorization, cloze and editing use only stanzas that have never been published. They come from two sources: stanzas the second author wrote as secondary-school coursework (*student* stanzas), used verbatim (we supply only their syllable segmentation), and stanzas composed by Claude Opus 4.7, screened for syllabification errors and invented words. Every stanza must satisfy all rules of its form under its syllable segmentation (meter score 1.0). Of the student stanzas, 2 of 10 *Khlong* (the others misplace tone marks), 13 of 14 *Klon* and 15 of 20 *Kap* qualify; none fits the *Chan* template. Per form we take up to 12, 14, 20 and 12 stanzas (*Khlong*, *Klon*, *Kap*, *Chan*), student stanzas first, giving 51 stanzas: 30 student-written (2/13/15/0) and 21 model-composed (10/1/5/5). Each stanza yields up to four cloze items blanking different positions (six for *Chan*, the smallest pool) and up to four editing items with the fault in different slots (every light slot for *Chan*); a fault is kept only if it breaks the meter. This gives 160 cloze items (48/42/40/30) and 74 editing items (48 *Khlong*, 26 *Chan*). Seven further model-composed stanzas (*Klon* 4, *Kap* 3; 18 cloze items) are used only to compare the two sources.

Scoring. Outputs are cleaned as in composition; if an output has more lines than the stanza (e.g. a verse preface explaining the fix), we grade the window of lines that best matches the prompt. The returned stanza is scored with the meter grader of Section 4.1 (same weights, best of the two layouts). Syllables the output shares with the prompt keep the segmentation we supply for the source stanza; new material (a fill or an edited stretch) is segmented with the dictionary tokenizer of Section 4.1. A faithful restoration of the source thus scores exactly 1.0. Segmenting whole outputs with the tokenizer instead lowers solve rates (GPT-5.5 cloze: $68 \rightarrow 47\%$) because it mis-splits retained source words; it changes the order of models only between models less than one point apart. An output is *valid* if its meter score is 1.0, and *solved* if it is valid and faithful to the prompt: in cloze, every given syllable is kept and every blank is filled; in editing, every line other than the faulty one is unchanged. A non-delivery counts as neither. Items built from the same stanza are correlated, so 95% confidence intervals come from a cluster bootstrap over source stanzas (2,000 resamples).

	Composition		Cloze: solved (%)					Editing: solved (%)			Edit.
Model	meter	valid	*Khlong*	*Klon*	*Kap*	*Chan*	All [95% CI]	*Khlong*	*Chan*	All [95% CI]	valid
items	160	160	48	42	40	30	160	48	26	74	74
GPT-5.5 (medium)	**0.896**	**58.1**	**52**	95	78	40	**68** [59,77]	**21**	**15**	**19** [9,30]	**50**
Gemini 3.5 Flash	0.820	38.1	33	**98**	**85**	**53**	67 [58,76]	2	4	3 [0,7]	20
Gemma 4 31B	0.729	25.0	15	64	30	27	34 [26,42]	2	0	1 [0,4]	9
DeepSeek V4 Flash	0.415	3.8	8	60	50	13	33 [24,43]	0	0	0 [0,0]	1
Typhoon-S 8B TH	0.258	0.0	6	31	18	13	17 [11,23]	**21**	0	14 [4,26]	14
OpenThaiGPT 8B TH	0.246	0.0	8	26	8	3	12 [7,18]	10	0	7 [0,15]	8
Qwen3-8B	0.233	0.6	8	19	12	10	12 [8,17]	2	0	1 [0,4]	3
Typhoon 2.5 30B TH	0.219	0.0	2	31	10	3	12 [7,18]	4	0	3 [0,9]	3

Table 12: Cloze and editing results (% of items). *Composition*: the model's meter score and share of fully valid stanzas on the composition task (Table 3). *Solved* = the returned stanza satisfies every rule of its form (meter score 1.0) and is faithful to the prompt; *Edit. valid* = meter score 1.0 regardless of changes elsewhere in the stanza. Brackets: 95% cluster-bootstrap CIs over source stanzas. A non-delivery (token budget exhausted) counts as unsolved; it mainly affects DeepSeek V4 Flash (24% of cloze and 50% of editing items). Bold: best in each column. TH = Thai-trained.

Findings. Cloze follows the composition ranking up to near-ties: GPT-5.5 and Gemini solve 67–68% of cloze items, Gemma and DeepSeek 33–34%, and the Thai-trained models and Qwen3-8B 12–17% (Table 12). Cloze is easier than composition for every model: GPT-5.5 solves 68% of cloze items but composes a fully valid stanza 58% of the time, and the four 8B–30B models, which almost never compose one, solve 12–17%. Difficulty follows the forms as in composition: *Klon* is the easiest form for every model, while *Khlong* and *Chan* stay hard (8–53% for the four stronger models); within *Chan*, these models satisfy 86% of blanked heavy slots but only 28% of light slots. Editing is the hardest mode and reorders the models. The four stronger models restore the missing *Khlong* tone mark in 65% of delivered outputs, but they also rewrite lines that had no error: GPT-5.5 returns a valid stanza for 50% of editing items yet solves 19%, and Gemini 20% versus 3%. Every valid Typhoon-S edit is also minimal, so it matches GPT-5.5 on *Khlong* (21%). A heavy syllable in a *Chan* light slot is rarely repaired (20% of delivered outputs for the four stronger models, 3% for the Thai-trained ones). The source of the stanza does not change this picture: in cloze, the four stronger models do better on student-written *Klon* (79 vs. 68%) and worse on student-written *Kap* (58 vs. 72%) and *Khlong* (25 vs. 28%) than on model-composed stanzas; in editing they rewrote the model-composed *Khlong* stanzas more often (solved 4 vs. 19%), a comparison that rests on two student stanzas.