PACUTE: Phonology-, Affix-, and Character-level Understanding of Tokens for Filipino

Anonymous ACL submission

Abstract

Large language models (LLMs) process text as sequences of subword tokens, which can obscure the character-level and morphological structure that underlies word formation. This limitation is most acute for languages with non-concatenative morphology, where standard tokenizers systematically misalign token boundaries with morpheme boundaries. We introduce PACUTE, a diagnostic benchmark of 4,600 tasks designed to evaluate morphological understanding in Filipino, a language characterized by productive infixation, reduplication, and diacritic-driven lexical distinctions that are typically absent from written text. PACUTE includes a hierarchical diagnostic framework of six compositional levels that localizes where morphological understanding breaks down. Evaluating open-weight LLMs and frontier commercial models, we find that open-weight models perform near chance on morpheme decomposition regardless of scale. Frontier models perform much better, often recovering individual affixes under contains-match scoring, but remain far below their character-level ceilings on compositional tasks of morpheme transformations and syllabification. These results identify productive morphological composition, rather than character access alone, as the persistent bottleneck for Filipino word-structure understanding.

1 Introduction

Large language models segment text into subword tokens and process them as atomic units, which limits their direct access to character-level structure. Benchmarks such as CUTE (Edman et al., 2024) and LangGame (Sims et al., 2025) demonstrate that many LLMs struggle with basic token-level tasks such as counting characters, character manipulations, and substring detection.

Filipino poses distinct challenges for tokenizers. We claim that Filipino, the national language of the Philippines, is a natural (and naturalistic) testbed to evaluate such capabilities in LLMs, for two reasons.

Productive non-concatenative morphology. Filipino uses infixes (e.g., *-um-*, *-in-*) that split root words and partial reduplication of initial syllables. For example, the root *kain* "eat" becomes *kumain* "ate" via the actor-focus infix *-um-*, inserted after the first consonant. Standard byte-pair encoding tokenizers (Sennrich et al., 2016) are suboptimal for language model pre-training (Bostrom and Durrett, 2020), and frequently segment such words at arbitrary boundaries that ignore morpheme structure.

Omitted stress and glottal markers. Filipino orthography can encode lexical distinctions through diacritics: acute (pahilís), grave (paiwà), and circumflex (pakupyâ). These create minimal pairs (Warstadt et al., 2020): *bása* "read" vs. *basâ* "wet"; *súka* "vomit" vs. *sukà* "vinegar"; *táyo* "we" vs. *tayô* "stand". However, everyday Filipino text typically omits these markers, creating systematic lexical ambiguity invisible to models.

Based on this, we introduce PACUTE (**P**honology-, **A**ffix-, and **C**haracter-level **U**nderstanding of **T**okens **E**valuation), a benchmark of 4,600 synthetic tasks targeting Filipino composition, manipulation, morphological extraction, morphological production, and syllabification, each available in multiple-choice (log-probability) and generative formats. PACUTE also includes a hierarchical diagnostic set (six levels) that separates prerequisite character skills from morpheme-level operations, enabling localization of bottlenecks.

We evaluate several pre-trained and instruction-tuned LLMs across major open families and sizes, as well as run continued pre-training on Gemma-2-2B with three preprocessing/tokenization regimes: vanilla BPE, stochastic token expansion (Stochas-

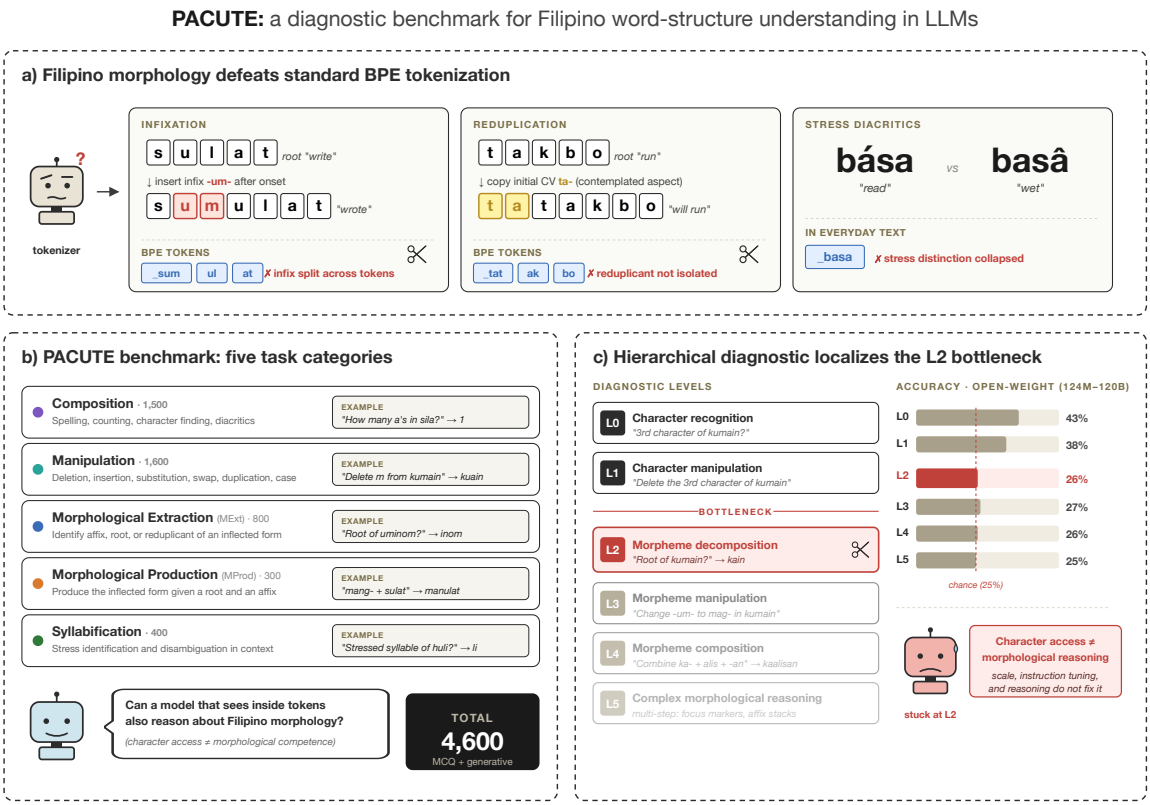

Figure 1: **Overview of PACUTE.** (A) Filipino morphology poses challenges for standard tokenizers: infixes split roots, reduplication copies syllables, and stress diacritics are typically omitted in text. (B) PACUTE comprises four task categories targeting different levels of word structure understanding. (C) A six-level hierarchical diagnostic localizes failures: models show above-chance performance on character-level tasks (L0–L1) but collapse to chance on morpheme decomposition (L2), with downstream levels inheriting this failure regardless of scale.

Tok) (Sims et al., 2025), and a Filipino morphology-aware expand/contract method based on Stochas-Tok. Our results show: **(i) open-weight models collapse to chance on morpheme decomposition** under MCQ log-probability scoring, despite above-chance performance on lower-level character tasks; and **(ii) frontier commercial models** often recover individual affixes, but **remain substantially below their character-level ceilings on hierarchical morpheme transformations and syllabification**; Filipino-relevant pre-training improves mid-range models, while tokenization interventions alone produce limited gains and can induce catastrophic forgetting.

2 Related Work

Relevant properties and evaluation of Filipino. Filipino has been characterized in the linguistic literature as exhibiting several non-concatenative morphological processes, including infixation (e.g., *-um-*, *-in-*), partial and full reduplication, and distinctions in stress and glottal stop that may be encoded through diacritics (Schachter and Otanes, 1983; Yap, 1967). These phenomena interact closely with phonological structure and are not always transparently reflected in surface orthography, particularly in naturally occurring text where diacritics are frequently omitted. As a result, many linguistically meaningful contrasts in Filipino are obscured at the character and token levels typically used in computational models.

Despite these properties, computational research on Filipino has largely focused on downstream NLP tasks such as sentiment analysis, part-of-speech tagging, and named entity recognition (Cruz and Cheng, 2022; Visperas et al., 2023). A recent benchmark, BATAYAN (Montalan et al., 2025), broadens evaluation coverage across understanding and generation tasks, yet remains primarily task-oriented rather than diagnostic of Filipino-specific linguistic structure.

Token and character level evaluations. In the broader literature, several studies have introduced token- and character-level diagnostic evaluations, for other languages. These include tests based on spelling, orthographic similarity, and character-level perturbations (Itzhak and Levy, 2022; Huang et al., 2023), as well as synthetic tasks involving letter counting, prefix or suffix identification, and substring detection (Kaushal and Mahowald, 2022; Efrat et al., 2023). CUTE extends this line of work by systematically evaluating models' token composition knowledge through character-level manipulations and token-based reasoning tasks (Edman et al., 2024). While effective as a general diagnostic framework, CUTE is designed to be language-agnostic and does not explicitly target non-concatenative morphology or phonology-driven alternations.

Morphology benchmarks such as SIGMORPHON evaluate inflection across diverse languages (Vylomova et al., 2020), but emphasize concatenative processes and transparent orthography, which are assumptions that do not hold for Filipino. Subword tokenization variants such as BPE-dropout (Provilkov et al., 2020) improve robustness to surface variation, but evaluation has focused on downstream performance rather than linguistically grounded morphological structure. PACUTE fills this gap with Filipino-specific diagnostics targeting phonology, affixation, and character-level structure.

3 PACUTE: Task Suite

Goal. PACUTE is a diagnostic benchmark for Filipino word-structure competence: whether a model can (i) access character-level information inside tokens and (ii) use that information to reason about productive morphology (especially infixation and reduplication). PACUTE is not designed mainly to measure general Filipino "understanding" via downstream tasks; rather, it targets concrete, local operations that underlie morphological generalization.

Why tokenization is a plausible failure source. Standard subword tokenizers (e.g., BPE) optimize compression and frequency statistics rather than morpheme boundary preservation. For Filipino, infixes and reduplicated segments often fall inside token interiors or are split inconsistently across tokens, making them hard to access as reusable units during pre-training and inference. Diacritics encoding lexical stress distinctions (*bása* "read" vs. *basâ* "wet") are nearly always absent from digital text (Almario, 2014), so models face a systematic mismatch between lexicographic and naturally occurring forms. Together, infixation, reduplication, morphophonemic alternations, and diacritic omission create a setting where surface-form competence is not sufficient for morphological competence (see Appendix A for a detailed treatment). This motivates two PACUTE design choices: (i) tasks that explicitly require operations aligned with morphological structure rather than only generic character probes, and (ii) a hierarchical diagnostic that separates prerequisite character skills from morpheme-level skills, allowing localization of where the compositional pipeline breaks down.

Task formats. Each PACUTE task is provided in two evaluation formats. **MCQ** instances present four answer options (chance = 25%). **GEN** instances require the model to produce an answer string. We include both formats because MCQ reduces generation and formatting confounds, while GEN tests whether the model can reliably execute the operation. Scoring procedures for both formats are described in §5.

3.1 Main suite and coverage

The main PACUTE suite contains 4,600 tasks across five categories, each targeting a different slice of subtoken competence: (i) **Composition** (950 MCQ + 550 GEN), (ii) **Manipulation** (800 MCQ + 800 GEN), (iii) **Morphological extraction** (400 MCQ + 400 GEN), (iv) **Morphological production** (150 MCQ + 150 GEN), (v) **Syllabification** (200 MCQ + 200 GEN). The composition and manipulation tasks provide character-level controls; the morphological extraction, morphological production, and syllabification tasks target Filipino-specific structure.

Composition tasks. Composition tasks test whether a model can use orthographic information that is typically hidden by tokenization: character counting, identification of specific characters/diacritics, and simple string properties. These are Filipino-adapted variants of standard character-level probes, using Filipino lexicon and (where relevant) diacritic-bearing forms. The intent is to separate "can the model see inside tokens?" from morphology-specific reasoning.

Manipulation tasks. Manipulation tasks test deterministic character operations such as insertion, deletion, substitution, and swapping. These tasks are intentionally low-level: they test whether models can execute controlled edits that mirror the mechanics needed for infix insertion and reduplication (even if the correct position for the edit is linguistically determined in the affixation tasks). This category also functions as a sanity check: if a model cannot reliably delete or insert characters, failures on morphology may be uninterpretable.

Morphological extraction and production tasks. Morphological tasks test whether models can identify and apply Filipino affixes, including prefixes, suffixes, and crucially, infixes. *Extraction* asks which affix, reduplicant, or root can be identified from an inflected form (e.g., word *kumain* → root *kain* + completed aspect infix *-um-*). *Production* asks for the correct derived form given a root and affix (e.g., apply completed aspect affix *-in-* to *sulat* → *sinulat*).

Syllabification tasks. Syllabification tasks test the phonological scaffolding needed for Filipino morphology. We treat syllable-level competence as an intermediate level between raw character manipulation and morpheme-level reasoning. This category includes stress identification tasks, which ask the model to identify the stressed syllable of a given word as it appears in a disambiguating sentence context. Because Filipino orthography conventionally omits stress diacritics, stress assignment is non-trivial: many words are orthographically identical but phonologically and semantically distinct depending on which syllable carries primary stress (e.g., *sála* "sin" vs. *salà* "filter" vs. *salâ* "broken"). Models must therefore use both lexical knowledge and sentential context to resolve stress.

Hierarchical diagnostic benchmark. In addition to the main suite, PACUTE includes a hierarchical diagnostic benchmark (600 MCQ + 600 GEN) organized into six levels: Level 0 (character recognition), Level 1 (character manipulation), Level 2 (morpheme decomposition), Level 3 (morpheme manipulation), Level 4 (morpheme composition), Level 5 (complex multi-step transformations). The hierarchy is designed to localize failures: Levels 3–5 presuppose that the model can decompose a word into morphemes (Level 2). Interpretation of per-level results is discussed in §5.

Table 1: PACUTE task statistics by category and format. *LangGame* = language-agnostic control; *MDA (multi-digit addition)* = non-linguistic control; *CUTE* = character-understanding control (generative only).

Category	MCQ	GEN	Total
Main suite			
Composition	950	550	1,500
Manipulation	800	800	1,600
Morph. extraction	400	400	800
Morph. production	150	150	300
Syllabification	200	200	400
Hierarchical (L0–L5, 100 MCQ+GEN each)			
Total hierarchical	600	600	1,200
Controls			
LangGame / MDA	1,000	1,000	2,000
CUTE	—	1,400	1,400
Total			**9,200**

Prompting and instance structure. All instances use short, uniform templates with minimal instruction overhead. GEN instances use a structured XML format with a `<reflection>` reasoning trace followed by an `<answer>` block; the evaluator extracts only the answer before scoring. Full prompting details, chat-template injection, and thinking-mode handling are described in Appendix B.

4 Data Construction

All PACUTE tasks are generated deterministically from public lexical resources and hand-curated annotations, without any use of generative models for content production.[1] This design choice is motivated by two considerations: (i) avoiding contamination of the benchmark with patterns already present in LLM pre-training corpora, and (ii) ensuring that every instance has a verifiable gold answer derived from linguistic rules rather than model-generated text (Bender et al., 2021).

Lexical resources. The primary source is a syllabified word list from the *UP Diksiyonaryong Filipino* (Almario, 2001) (16,828 entries with syllable boundaries, stress, and POS), paired with a Filipino corpus frequency list (118,801 forms) for frequency-weighted sampling. Morphological tasks draw from a manually curated dataset of inflection (e.g., *takbo* + *-um-* → *tumakbo*), assimilation (e.g., *naNG-* + *bigay* → *namigay*),

[1] Code and benchmark data will be made publicly available upon acceptance.

and reduplication patterns (Zamar, 2022; Schachter and Otanes, 1983). Syllabification tasks use a handcrafted corpus of minimally contrastive word pairs in disambiguating sentence context (e.g., *búhay* "life" vs. *buháy* "alive"). All task instances are generated by a deterministic Python pipeline; items are available in English and Filipino prompt formats and summarized in Table 1.

5 Evaluation

Overview. We use PACUTE to evaluate (i) a broad set of LLMs (zero-shot) and (ii) a controlled continued-pre-training (CPT) study where we vary tokenization during Filipino-domain adaptation. We report normalized accuracy and F1 score on MCQ tasks (4-way; chance = 25%) and string-match accuracy on generative tasks.

Models. We evaluate open-weight models spanning several major families (Gemma, GPT-2, GPT-OSS, Llama, Mistral, Phi, Qwen, SEA-LION, DeepSeek, Kimi), parameter scales from 124M to 1T, and both base pre-trained, instruction-tuned, and thinking variants. We additionally evaluate frontier commercial models (Claude 4 series, GPT-5 series, and Gemini 3 series) on generative benchmarks only (commercial APIs for most models do not expose token log-probabilities for MCQ scoring). This mix allows us to separate effects of scale, instruction tuning, regional pre-training data (SEA-LION), and frontier capability.

Scoring. All models are evaluated on the full PACUTE suite, hierarchical benchmark, and controls (CUTE, LangGame, MDA). For MCQ, we select the option with highest conditional log-probability and report normalized accuracy and F1 (4-way; chance = 25%). For GEN, we use exact match after lightweight normalization (case, whitespace, leading/trailing hyphens on affix strings) and contains-match for tasks where models may produce extra context. Per-level hierarchical results (L0–L5) are interpreted diagnostically: if L2 (morpheme decomposition) is at chance, L3–L5 are expected to collapse regardless of L0–L1 performance.

Continued pre-training (CPT) setup. For CPT, we adapt Gemma-2-2B on the SEA-PILE v2 Filipino corpus (7.4GB) for up to 5,000 steps, saving checkpoints every 1,000 steps. We compare three segmentation regimes applied at preprocessing time while keeping the underlying model architecture fixed: (i) vanilla (unchanged BPE tokenization), (ii) StochasTok (stochastic token expansion; $\sim$10% split rate), (iii) Patok (morphology-aware expand/contract using Filipino affix rules). We evaluate each checkpoint on the same benchmark suite.

Human baseline. We establish a human baseline on a PACUTE subset using three native Filipino speakers, reporting average accuracy and Fleiss' κ for inter-annotator agreement (Appendix D).

6 Results

Full numerical results are reported in Tables 5, 6, 7, 8, 9, 10, 11, and in the Appendix. Figure 2 summarizes the main pattern: strong models approach ceiling on character-level and language-agnostic controls, but remain substantially weaker on Filipino-specific phonological and morphological tasks. The gap is clearest on the Hierarchical benchmark and on Syllabification, which require models to use morpheme structure, syllable structure, stress, and phonologically conditioned transformations rather than simple character edits.

Morpheme decomposition is difficult for open-weight models. Among open-weight models evaluated with MCQ log-probability scoring, performance on the Hierarchical benchmark remains close to chance. Normalized accuracy ranges from -3.6 to $+2.9$ across 32 open-weight variants (mean: -0.6). This collapse is driven by Level 2 morpheme decomposition: models often fail to identify the root and affix structure of Filipino words, even when they perform above chance on lower-level character recognition and manipulation tasks. Scale alone does not remove this failure. Larger open-weight models, including Gemma-4-31B and Qwen-3.6-27B, do not reliably outperform much smaller models on morpheme decomposition.

Character-level controls are largely solved by frontier models. Commercial models perform near ceiling on the easiest controls: MDA (99.4–100.0% CM), LangGame (94.9–100.0%), and Manipulation (GPT-5.5: 100.0%, DeepSeek-R1: 96.9% CM). These results confirm that PACUTE's harder tasks reflect genuine morphological difficulty, not generic instruction-following failure.

Morphological extraction is recoverable for strong models, but remains formatting-sensitive.

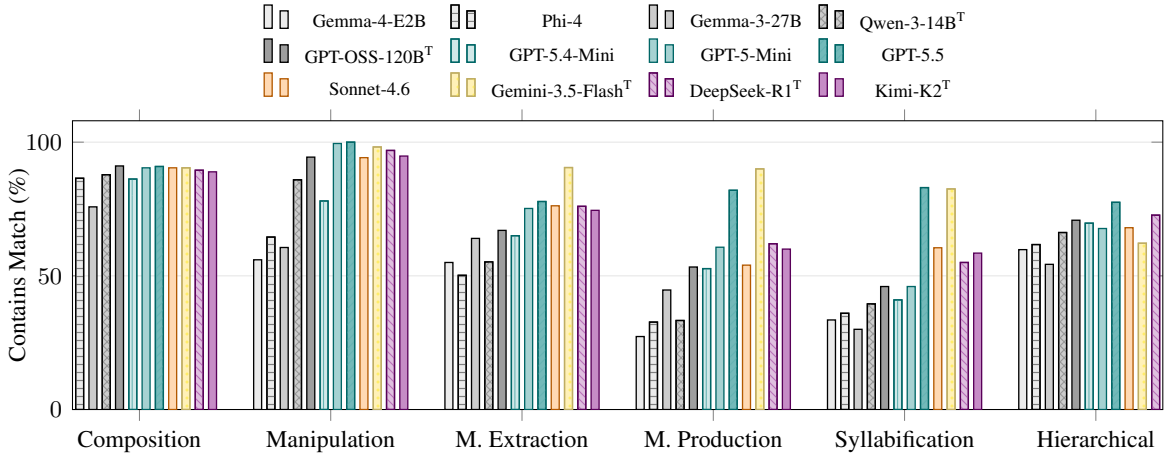

Figure 2: **Contains-match accuracy across PACUTE categories** for 12 models spanning open-weight (gray, 5B–120B) and frontier commercial (color) tiers. Superscript ^{T} = extended thinking/reasoning. Composition and Manipulation approach ceiling for strong models, while **Hierarchical and Syllabification remain suppressed across all classes**. Even GPT-5.5 (77.5% Hierarchical-CM) sits 22pp below its Manipulation ceiling. DeepSeek-R1 leads Hierarchical (72.7% CM); Gemini-3.5-Flash dominates Morphological Extraction and Production (90.5/90.0% CM) but flops on Hierarchical (62.2%), suggesting morpheme operations and multi-step composition recruit partially disjoint capabilities.

Performance on Morphological Extraction is consistently higher under contains-match than exact match (e.g., Gemini-3.5-Flash: 89.5% EM / 90.5% CM; Claude-Sonnet-4.6: 74.0% / 76.2%), indicating models often recover the correct affix but not always in the required form.

Hierarchical transformations remain the central bottleneck. The Hierarchical benchmark remains substantially below the character-level ceiling. The best model, GPT-5.5, reaches 64.7% EM and 77.5% CM on Hierarchical-GEN, compared with 100.0% CM on Manipulation-GEN. DeepSeek-R1 reaches 72.7% CM, Kimi-K2-Thinking reaches 67.2%, Claude Sonnet 4.6 reaches 68.0%, and GPT-5-Mini reaches 67.7%. This pattern suggests that the difficulty is not identifying individual characters or even recovering individual affixes, but composing several operations: decomposing a word into morphemes, manipulating those morphemes, and recombining them into the correct surface form. PACUTE therefore isolates a gap between local subtoken access and productive morphological reasoning.

Syllabification exposes a separate phonological weakness. Syllabification is difficult even for frontier models. The best performers are GPT-5.5 (83.0% CM) and Gemini-3.5-Flash (82.5%), while several strong models score much lower (GPT-5-Mini: 46.0%; Gemini-3.1-Flash-Lite: 38.0%). Even models that handle character manipulation well may still lack stable representations of Filipino syllable structure, stress, and the *ng* digraph.

Reasoning models narrow but do not close the gap. Thinking models are often strong, but their gains are uneven. DeepSeek-R1 obtains one of the highest Hierarchical scores among commercial models (72.7% CM), and Kimi-K2-Thinking also performs competitively (67.2% CM). Gemini-3.5-Flash is especially strong on Morphological Extraction, Morphological Production, and Syllabification, reaching 90.5%, 90.0%, and 82.5% CM respectively, but remains lower on Hierarchical at 62.2% CM. These results suggest that reasoning can help when the relevant linguistic knowledge is available, but it does not by itself guarantee robust morpheme decomposition or composition.

Pre-training data matters below the frontier. Among open-weight models, SEA-LION models trained on Southeast Asian data outperform generic models of comparable scale on Filipino-specific tasks. SEA-LION-Qwen-v4.5-27B achieves 85.8 normalized accuracy on MProd-MCQ and 27.5 on Manip-MCQ, compared with 83.1 and 26.2 for Qwen-3.6-27B. This suggests that Filipino-relevant pre-training data improves mid-range model performance, even though frontier commercial models can partially compensate through general

capability and stronger instruction following.

Human baseline. Human performance is substantially above model performance on all categories. Humans particularly outperform the strongest models on MExt-MCQ (0.925 vs. 0.895 EM, MProd-GEN (0.911 vs. 0.900 EM), and Syll-GEN (0.967 vs. 0.830 EM). Overall agreement among annotators for PACUTE is high (90.4% for GEN, 85.0% for MCQ). Moreover, Fleiss' κ of 85.5% for MCQ and 74.0% for GEN indicates almost perfect and substantial agreement respectively.

Summary. PACUTE reveals a hierarchy of competence: strong models solve character edits and language-agnostic controls, and often recover individual affixes, but remain well below their character-level ceilings on Hierarchical and Syllabification tasks. Scale, instruction tuning, and reasoning all help, but productive morphological composition (such as decomposing words, tracking syllable structure, applying phonologically conditioned transformations) remains the persistent bottleneck.

7 Error Analysis

We conduct a manual error analysis on sampled GEN responses from instruction-tuned models, focusing on GEN because reasoning traces and surface-form choices are visible there. We identify four recurring error types.

Type I: Instruction following failure. Many GEN errors reflect non-compliance rather than a linguistic gap: generating a full sentence instead of a word, explaining instead of answering, or responding in English to a Filipino prompt. Thinking-mode models are especially prone, as the correct answer often appears mid-chain but is not surfaced as the final response. These errors inflate raw GEN error rates and motivate contains-match scoring.

Type II: Linguistic error. Models attempt the correct operation but apply rules incorrectly. The most frequent subtype is infix placement error: the infix is identified (e.g., *-um-*) but inserted at the wrong position (e.g., **sulumat* instead of *sumulat*). Other subtypes include root boundary confusion, *ng*-digraph mishandling in syllabification, and reduplication copying at the wrong edge.

Type III: Reasoning inconsistency. Thinking models (e.g., Qwen-3 variants) sometimes reach the correct intermediate conclusion in the scratchpad yet output a contradicting final answer, such as by correctly counting syllables then rationalizing a different number, or identifying the right root/affix then selecting a wrong MCQ option. This suggests the final answer is not robustly conditioned on the chain-of-thought.

Type IV: Systematic model bias. The most distinctive error class is a **penultimate-stress bias**: models default to labeling the second-to-last syllable as stressed regardless of actual diacritic position. Analogous biases appear in affix selection (preference for high-frequency *mag-* even in illicit contexts). Neither scale nor extended reasoning corrects these biases. Models also fail to use sentential context for stress disambiguation—e.g., Gemma-3-27B-IT answers *ga* for *galing* in all contexts even when the sentence selects the noun reading requiring final stress—and treat the digraph *ng* as two Unicode characters, causing systematic syllable overcounting.

8 Discussion

Morpheme decomposition as a training data problem. The near-random performance on Level 2 decomposition tasks among open-weight models points to a missing-knowledge failure rather than an insufficient-capacity failure. Frontier commercial models partially overcome this floor, but their residual errors concentrate specifically on phonologically conditioned processes—nasal assimilation, vowel-initial root alternations—that require productive rule application rather than memorization. This suggests two stacked bottlenecks: (i) Filipino morphological structure is underrepresented in pre-training corpora of English-dominant models, and (ii) even when models acquire partial affix knowledge at scale, they fail to learn the phonological rules governing affix–root interactions. BPE tokenization compounds both by fragmenting infixed and reduplicated forms at boundaries that obscure morpheme structure. Composition tasks probing diacritic identification score notably lower than non-diacritic character counting for most models, consistent with stress and glottal markers being nearly absent from everyday Filipino text (Almario, 2014)—a gap with downstream consequences for stress disambiguation and lexical sense selection

invisible to task-oriented benchmarks.

Why tokenization-based interventions have limited impact. The CPT results with StochasTok and Patok (Appendix E) show that all three regimes, including vanilla BPE, underperform the un-CPT'd base on the PACUTE main-suite. Crucially, Hierarchical accuracy remains pinned at the chance floor for all three regimes (21.2–24.7%), confirming that L2 morpheme decomposition is unmoved by either stochastic token expansion or morphology-aware segmentation. This is consistent with prior work showing that BPE-dropout and similar methods improve robustness to surface variation but do not inject morphological abstraction when the underlying training signal is absent (Bostrom and Durrett, 2020; Sims et al., 2025). None of these CPT setups yield a net gain for Filipino morphology, and simply exposing the model to more Filipino text under alternative segmentation schemes is insufficient without mechanisms that preserve general capabilities or explicitly align token boundaries with morpheme boundaries at scale.

Implications for low-resource morphologically complex languages. PACUTE's diagnostic hierarchy provides a template for identifying where model competence breaks down for languages with non-concatenative morphology. The L0/L1 scores show that models can access character-level information inside tokens for Filipino to a limited degree; the abrupt collapse at L2 precisely locates the gap. This has practical implications: downstream Filipino NLP tasks that rely on aspect or voice distinctions (which are morphologically encoded) should not be expected to benefit from scale alone if morpheme decomposition is absent. The SEA-LION advantage suggests that domain-specific pre-training on high-quality Filipino text is the most actionable intervention currently available.

Limitations

PACUTE evaluates word-level morphological and character-level competence in a controlled, synthetic setting. Several limitations should be noted. All tasks are generated from a curated lexical resource and a manually annotated affix set. The word pool (16,828 entries from the *UP Diksiyonaryong Filipino*) does not cover informal registers, code-switching, or borrowings from Spanish and English, which are pervasive in natural Filipino text; infix instances in particular are underrepresented in the affixation task set, limiting the diagnostic power of infix-specific categories in the hierarchical benchmark. Model performance on PACUTE may therefore not transfer directly to naturalistic Filipino NLP tasks. Tasks that involve diacritic-bearing forms assume that models have been exposed to or can infer stress distinctions. However, diacritics are nearly always absent from digital Filipino text, meaning that performance on diacritic tasks may underestimate competence that a model could demonstrate if given diacritic-marked input.

Ethical Considerations

All lexical data are drawn from published reference materials, namely the syllabified word list from the *UP Diksiyonaryong Filipino* (Almario, 2001), morphological patterns from (Zamar, 2022) and (Schachter and Otanes, 1983), and from the SEA-PILE-v2 Filipino text corpus (Ng et al., 2025). No personally identifiable information, social media data, or sensitive personal communications were used in task construction. SEA-PILE v2 is released under the ODC-By 1.0 license; we use it in accordance with its terms. The *UP Diksiyonaryong Filipino* is a copyrighted reference work published by the University of the Philippines. PACUTE is intended as a diagnostic tool for identifying specific morphological competence gaps in LLMs, not as a general-purpose measure of Filipino language ability. PACUTE should not be used to rank models for production deployment in Filipino-language applications without considering downstream task performance. Filipino is the national language of the Philippines. We acknowledge that the computational treatment of Filipino morphology in this work is primarily from a formal linguistic perspective and may not capture the full range of dialectal variation or the sociolinguistic complexity of language use in the Philippines. We encourage follow-up work that extends PACUTE-style diagnostics to other Philippine languages (e.g., Cebuano, Ilocano) and to more naturalistic evaluation settings.

Use of AI assistants. AI language models (Claude) were used for proofreading and editing of manuscript text. No AI-generated content was used for research design, data construction, or analysis.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Virgilio S. Almario. 2001. *UP Diksiyonaryong Filipino*. Sentro ng Wikang Filipino, Anvil Publishing, Manila, Philippines.

Virgilio S. Almario. 2014. *KWF Manwal sa Masinop na Pagsulat*. Komisyon sa Wikang Filipino.

Anthropic. 2025. System Card: Claude Haiku 4.6.

Anthropic. 2026a. System Card: Claude Opus 4.6.

Anthropic. 2026b. System Card: Claude Sonnet 4.6.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. *arXiv preprint arXiv:2004.03720*.

Jan Christian Blaise Cruz and Charibeth Cheng. 2022. Improving large-scale language models and resources for filipino. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6548–6555.

Google DeepMind. 2026. Gemma 4: Open models for advanced reasoning and agentic workflows. `https://ai.google.dev/gemma/docs/core`.

DeepSeek-AI. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI. 2025b. Deepseek-v3.2: Pushing the frontier of open large language models. *Preprint*, arXiv:2512.02556.

Lukas Edman, Helmut Schmid, and Alexander Fraser. 2024. CUTE: Measuring LLMs' understanding of their tokens. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3017–3026, Miami, Florida, USA. Association for Computational Linguistics.

Avia Efrat, Or Honovich, and Omer Levy. 2023. Lmentry: A language model benchmark of elementary language tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10476–10501.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Google. 2026a. Gemini 3.5: frontier intelligence with action. Large language model.

Google. 2026b. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-flash-lite/. Large language model.

Jing Huang, Zhengxuan Wu, Kyle Mahowald, and Christopher Potts. 2023. Inducing character-level structure in subword-based language models with type-level interchange intervention training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12163–12180.

Itay Itzhak and Omer Levy. 2022. Models in a spelling bee: Language models implicitly learn the character composition of tokens. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5061–5068.

Ayush Kaushal and Kyle Mahowald. 2022. What do tokens know about their characters and how do they know it? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2487–2507.

Microsoft, :, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi ling Chen, Qi Dai, and 57 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *Preprint*, arXiv:2503.01743.

Mistral. 2026. Introducing Mistral Small 4.

Jann Railey Montalan, Jimson Paulo Layacan, David Demitri Africa, Richell Isaiah S Flores, Michael T Lopez Ii, Theresa Denise Magsajo, Anjanette Cayabyab, and William Chandra Tjhi. 2025. Batayan: A filipino nlp benchmark for evaluating large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31239–31273.

Moonshot AI. 2025. Kimi K2: Open Agentic Intelligence. *Preprint*, arXiv:2507.20534.

Raymond Ng, Thanh Ngan Nguyen, Yuli Huang, Ngee Chia Tai, Wai Yi Leong, Wei Qi Leong, Xianbin Yong, Jian Gang Ngui, Yosephine Susanto, Nicholas Cheng, Hamsawardhini Rengarajan, Peerat

Limkonchotiwat, Adithya Venkatadri Hulagadri, Kok Wai Teng, Yeo Yeow Tong, Bryan Siow, Wei Yi Teo, Wayne Lau, Choon Meng Tan, and 12 others. 2025. Sea-lion: Southeast asian languages in one network. *Preprint*, arXiv:2504.05747.

OpenAI. 2025. Gpt-5-mini model card. Technical report, OpenAI, San Francisco, CA.

OpenAI. 2026a. Gpt-5.4-mini model card. Technical report, OpenAI, San Francisco, CA.

OpenAI. 2026b. Gpt-5.4-nano model card. Technical report, OpenAI, San Francisco, CA.

OpenAI. 2026c. Gpt-5.5 model card. Technical report, OpenAI, San Francisco, CA.

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, and 107 others. 2025. gpt-oss-120b and gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Qwen Team. 2026a. Qwen3.5: Towards native multimodal agents.

Qwen Team. 2026b. Qwen3.6-27B: Flagship-level coding in a 27B dense model.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Paul Schachter and Fe T Otanes. 1983. *Tagalog reference grammar*. Univ of California Press.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Anya Sims, Thom Foster, Klara Kaleb, Tuan-Duy H. Nguyen, Joseph Lee, Jakob N. Foerster, Yee Whye Teh, and Cong Lu. 2025. Stochastok: Improving fine-grained subword understanding in llms. *Preprint*, arXiv:2506.01687.

Moses L Visperas, Christalline Joie Borjal, Aunhel John M Adoptante, Danielle Shine R Abacial, Ma Miciella Decano, and Elmer C Peramo. 2023. itanong-ds: A collection of benchmark datasets for downstream natural language processing tasks on select philippine languages. In *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, pages 316–323.

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, and 1 others. 2020. Sigmorphon 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–39.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

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. *Preprint*, arXiv:2505.09388.

Fe Aldave Yap. 1967. *Synchronic analysis of tagalog phonemes*. Ph.D. thesis, University of British Columbia.

Maria Sheila Zamar. 2022. *Filipino: An essential grammar*. Routledge.

A Filipino Morphological Properties

We summarize the Filipino properties that motivate PACUTE's task design.

Affixation and non-concatenative processes. Filipino (Tagalog-based) word formation is highly productive and relies on a large inventory of affixes, including prefixes (e.g., *mag-*, *nag-*, *pag-*), suffixes (e.g., *-an*, *-in*), circumfixes (e.g., *pag-…-an*), and infixes (notably *-um-* and *-in-*) (Schachter and Otanes, 1983). Infixes are inserted inside the root rather than concatenated at an edge, typically but not always after the onset of the first syllable: *kain* → *k**um**ain*, *sulat* → *s**um**ulat* / *s**in**ulat*. This insertion position is sensitive to the phonological shape of the root (e.g., vowel-initial roots tend to surface with an infix realized prefixally), which complicates purely string-based decomposition.

Reduplication as a productive generator of novel forms. Filipino uses reduplication to mark grammatical aspect and related meanings (Zamar, 2022). Partial reduplication commonly copies the initial CV(C) of the root (*luto* → *luluto*, *takbo* → *tatakbo*), while full reduplication copies the entire root (*araw-araw*). Because these patterns are productive, they generate many surface forms that may be rare or unseen as wholes, even when the root is frequent.

Morphophonemic alternations at affix boundaries. Affixation in Filipino often triggers alternations that obscure a simple affix-root boundary (Schachter and Otanes, 1983). A canonical case is nasal assimilation in the *mang-* prefix, where the nasal assimilates to the place of articulation of the following consonant and may induce deletion or substitution (e.g., *mang-* + *bili* → *mamili*). More generally, attachment can involve phonologically conditioned changes that make decomposition a structured reasoning problem rather than a simple string-splitting task.

Stress, glottal stop, and diacritic omission. Filipino orthography can mark stress and glottal stop using diacritics (acute, grave, circumflex), yielding minimal pairs such as *básа* "read" vs. *basâ* "wet", *súka* "vomit" vs. *sukà* "vinegar," and *táyo* "we" vs. *tayô* "stand" (Yap, 1967). However, these diacritics are rarely written in contemporary digital text (Almario, 2014). For modeling, this creates a systematic mismatch between (i) lexicographic forms where distinctions are explicit and (ii) naturally occurring forms where distinctions are collapsed. As a result, tasks that require sensitivity to stress and glottal structure can fail either because the model lacks the concept or because the surface form underdetermines the target.

Implications for evaluation design. Taken together, infixation, reduplication, alternations, and diacritic omission create a setting where surface-form competence is not sufficient for morphological competence. A model may (i) recognize characters but fail to segment into morphemes, (ii) memorize frequent affixed words without learning productive rules, or (iii) handle concatenative affixes but fail on infixes because the relevant unit is non-contiguous. PACUTE is designed to disentangle these possibilities by pairing (a) character-level controls, (b) targeted affixation and syllabification tasks, and (c) a hierarchical framework that localizes errors to a specific compositional level.

B Prompting and Instance Structure

MCQ instances score options by model log-probability under a fixed prompt and require no special output format. GEN instances use a structured XML output format: the model produces a `<reflection>` block (1–2 sentences of reasoning) followed by an `<answer>` block containing only the target form. The evaluator extracts the content inside `<answer>` and discards the reflection before computing exact-match and contains-match scores. For instruction-tuned models, the format instruction is injected as the system prompt via the tokenizer's chat template; for base models, it is prepended as plain text. Thinking-mode models additionally produce a `<think>` block prior to the reflection, which is similarly stripped.

The reflection requirement serves two purposes: (i) it provides a lightweight, always-on reasoning trace enabling the qualitative error analysis in §7; and (ii) it encourages the model to commit to an explicit rationale before producing its answer, which is informative for diagnosing Type III errors (reasoning inconsistency). Prompts are intentionally simple and uniform—no few-shot examples, no chain-of-thought elicitation beyond the reflection format—to preserve comparability across base and instruction-tuned models.

C Model List

Tables 2 and 3 list all models evaluated in this work with parameter counts, variant types, and references. IT = instruction-tuned; PT = pretrained (base); Thinking = extended chain-of-thought mode.

Table 2: Base pre-trained models. Model names link to HuggingFace model pages.

Family	Model	Params	Reference
Gemma-4	google/gemma-4-26B-A4B	26B	Gemma Team et al. (2025)
	google/gemma-4-31B	31B	
	google/gemma-4-E2B	5B	
	google/gemma-4-E4B	8B	
GPT-2	openai-community/gpt2	124M	Radford et al. (2019)
	openai-community/gpt2-medium	355M	
	openai-community/gpt2-large	774M	
	openai-community/gpt2-xl	1.5B	
Qwen-3.5	Qwen/Qwen3.5-0.8B-Base	0.8B	Qwen Team (2026a)
	Qwen/Qwen3.5-2B-Base	2B	
	Qwen/Qwen3.5-4B-Base	4B	
	Qwen/Qwen3.5-9B-Base	9B	

Table 3: Instruction-tuned and reasoning variants. **IT** = standard instruction-tuned only; **IT + Thinking** = evaluated in both standard and thinking modes; **Thinking** = thinking-only model. Model names link to HuggingFace model pages.

Family	Model	Type	Params	Reference
Gemma-3	google/gemma-3-4b-it	IT	4B	Gemma Team et al. (2025)
	google/gemma-3-12b-it	IT	12B	
	google/gemma-3-27b-it	IT	27B	
Gemma-4	google/gemma-4-26B-A4B-it	IT + Thinking	26B	DeepMind (2026)
	google/gemma-4-31B-it	IT + Thinking	31B	
	google/gemma-4-E2B-it	IT + Thinking	5B	
	google/gemma-4-E4B-it	IT + Thinking	8B	
GPT-OSS	openai/gpt-oss-20b	Thinking	20B	OpenAI et al. (2025)
	openai/gpt-oss-120b	Thinking	120B	
Mistral-Small-4	mistralai/Mistral-Small-4-119B-2603	IT + Thinking	119B	Mistral (2026)
Phi-4	microsoft/Phi-4-mini-instruct	IT	3.8B	Microsoft et al. (2025)
	microsoft/Phi-4	IT	14B	Abdin et al. (2024)
Qwen-3	Qwen/Qwen3-4B	IT + Thinking	4B	Yang et al. (2025)
	Qwen/Qwen3-8B	IT + Thinking	8B	
	Qwen/Qwen3-14B	IT + Thinking	14B	
	Qwen/Qwen3-32B	IT + Thinking	32B	
Qwen-3.6	Qwen/Qwen3.6-27B	IT + Thinking	0.6B	Qwen Team (2026b)
SEA-LION-4	aisingapore/Gemma-SEA-LION-v4-27B-IT	IT + Thinking	27B	Ng et al. (2025)
	aisingapore/Qwen-SEA-LION-v4-32B-IT	IT + Thinking	32B	

Table 4: Commercial frontier models.

Provider	Model	Params	Reference
Anthropic	Opus 4.6	-	Anthropic (2026a)
	Sonnet 4.6	-	Anthropic (2026b)
	Haiku 4.5	-	Anthropic (2025)
DeepSeek	DeepSeek R1	685B	DeepSeek-AI (2025a)
	DeepSeek 3.2	685B	DeepSeek-AI (2025b)
Google	Gemini-3.5-Flash	-	Google (2026a)
	Gemini-3.1-Flash-Lite	-	Google (2026b)
MoonshootAI	Kimi K2	1T	Moonshot AI (2025)
OpenAI	GPT-5.5	-	OpenAI (2026c)
	GPT-5.4-Mini	-	OpenAI (2026a)
	GPT-5.4-Nano	-	OpenAI (2026b)
	GPT-5-Mini	-	OpenAI (2025)

D Evaluation Results

Table 5: Human baseline performance and inter-annotator agreement on the **MCQ** format of PACUTE (three annotators). **Mean Acc** is accuracy averaged across annotators ($\pm$std). **Raw Agree** is the fraction of items on which all three annotators selected the same option. **Fleiss** κ is computed over the four-class option labels ($\kappa \geq 0.81$ almost perfect; 0.61–0.80 substantial). **MCQ Avg** is the macro-average across all five benchmarks.

Benchmark	Mean Acc	Raw Agree	Fleiss κ
Composition	0.926 ± 0.03	90.5%	0.915
Manipulation	0.983 ± 0.01	98.8%	0.989
M. Extraction	0.942 ± 0.07	82.5%	0.842
M. Production	0.978 ± 0.03	93.3%	0.938
Syllabification	0.767 ± 0.14	60.0%	0.589
MCQ Avg.	**0.919**	**85.0%**	**0.855**

Table 6: Human baseline performance and inter-annotator agreement on the **GEN** format of PACUTE (three annotators). **Mean EM** is exact-match accuracy averaged across annotators ($\pm$std). **Raw Agree** is the fraction of items on which all three annotators produced identical normalized strings. **Fleiss** κ is computed on the binarized correct/incorrect judgment against the reference answer ($\kappa \geq 0.81$ almost perfect; 0.61–0.80 substantial). **GEN Avg** is the macro-average across all five benchmarks.

Benchmark	Mean EM	Raw Agree	Fleiss κ
Composition	0.982 ± 0.00	96.4%	1.000
Manipulation	1.000 ± 0.00	100.0%	1.000
M. Extraction	0.925 ± 0.04	75.0%	0.765
M. Production	0.911 ± 0.05	86.7%	0.451
Syllabification	0.967 ± 0.02	95.0%	0.483
GEN Avg.	**0.957**	**90.4%**	**0.740**

Table 7: Results for PT models on PACUTE benchmarks. All values are percentages.

Model	Comp-MCQ		Manip-MCQ		MExt-MCQ		MProd-MCQ		Syll-MCQ		Hier-MCQ		LGame-MCQ		MDA-MCQ	
	NormAcc	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*
Qwen/Qwen3.5-0.8B-Base	16.8	54.6	2.2	42.1	39.0	70.3	61.8	83.3	**-45.0**	**43.1**	0.2	40.2	19.2	56.5	51.9	78.0
Qwen/Qwen3.5-2B-Base	15.6	53.7	6.0	45.6	49.0	76.4	77.8	90.9	-48.0	41.3	-3.3	36.7	25.5	61.2	52.8	78.5
Qwen/Qwen3.5-4B-Base	17.2	55.0	9.5	48.6	60.3	82.5	85.8	94.4	-47.0	41.9	-2.0	38.1	52.0	78.0	58.0	81.3
Qwen/Qwen3.5-9B-Base	24.2	60.3	19.8	57.0	51.7	77.9	84.0	93.6	-50.0	40.0	-3.6	36.5	**68.0**	**86.4**	**59.5**	**82.1**
google/gemma-4-26B-A4B	16.8	54.6	**29.0**	**63.7**	61.0	82.9	89.3	95.8	-50.0	40.0	-0.9	39.1	55.6	80.0	58.0	81.3
google/gemma-4-31B	**27.4**	**62.6**	24.3	60.4	**63.7**	**84.2**	**91.1**	**96.6**	-48.0	41.3	-2.4	37.6	64.5	84.7	54.0	79.2
google/gemma-4-E2B	14.9	53.2	3.5	43.3	51.0	77.5	77.8	90.9	-52.0	38.7	0.0	40.0	22.1	58.8	52.0	78.0
google/gemma-4-E4B	16.8	54.6	15.7	53.7	56.7	80.6	87.6	95.1	-52.0	38.7	-0.7	39.4	37.2	69.2	48.8	76.2
openai-community/gpt2	8.9	48.1	-8.5	31.4	38.7	70.1	45.8	74.5	-51.0	39.4	1.8	41.7	5.5	45.1	0.7	40.6
openai-community/gpt2-large	11.0	49.9	-4.0	36.1	45.3	74.2	53.8	79.0	-52.0	38.7	2.7	42.5	2.7	42.5	-7.5	32.5
openai-community/gpt2-medium	14.5	52.8	-4.8	35.2	44.0	73.4	47.6	75.5	-52.0	38.7	**2.9**	**42.7**	5.9	45.4	-1.6	38.4
openai-community/gpt2-xl	9.9	49.0	-5.0	35.1	43.3	73.0	52.0	78.0	-49.0	40.6	2.2	42.1	4.5	44.2	-6.1	33.9

Comp = PACUTE Composition; *Hier* = Hierarchical; *LGame* = LangGame; *MDA* = Multi-Digit Addition; *MExt* = PACUTE Morphological Extraction; *MProd* = PACUTE Morphological Production; *Manip* = PACUTE Manipulation; *Syll* = PACUTE Syllabification; Bold = best in column. Cell colour: low → white → high per column.

Table 8: Results for IT models on PACUTE MCQ benchmarks. All values are percentages.

Model	Comp-MCQ		Manip-MCQ		MExt-MCQ		MProd-MCQ		Syll-MCQ		Hier-MCQ		LGame-MCQ		MDA-MCQ	
	NormAcc	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*	*NormAcc*	*F1*
Qwen/Qwen3-4B	18.9	56.3	8.5	47.8	48.3	76.0	84.9	94.0	-53.0	38.1	-3.1	37.0	47.7	75.6	57.7	81.2
Qwen/Qwen3-4B (thinking)	18.3	55.8	7.0	46.4	47.3	75.4	84.9	94.0	-53.0	38.1	-0.7	39.4	40.8	71.5	54.0	79.2
Qwen/Qwen3-8B	17.5	55.2	11.5	50.3	47.7	75.6	78.7	91.3	-44.0	43.7	-5.6	34.5	51.3	77.7	50.8	77.4
Qwen/Qwen3-8B (thinking)	17.5	55.2	11.3	50.2	47.0	75.2	78.7	91.3	-45.0	43.1	-4.9	35.2	51.7	77.9	50.9	77.5
Qwen/Qwen3-14B	17.3	55.1	16.5	54.4	52.7	78.4	84.9	94.0	-47.0	41.9	-3.3	36.7	67.3	86.0	52.9	78.6
Qwen/Qwen3-14B (thinking)	17.3	55.1	16.0	54.0	52.7	78.4	84.9	94.0	-47.0	41.9	-4.4	35.6	67.6	86.2	53.6	78.9
Qwen/Qwen3-32B	25.3	61.1	18.2	55.7	57.3	81.0	85.8	94.4	-52.0	38.7	-0.7	39.4	67.7	86.2	61.2	83.0
Qwen/Qwen3-32B (thinking)	25.5	61.2	18.2	55.7	57.0	80.8	85.8	94.4	-52.0	38.7	-0.9	39.1	67.5	86.1	61.1	82.9
Qwen/Qwen3.6-27B	24.6	60.6	26.2	61.7	62.3	83.6	83.1	93.2	-51.0	39.4	-3.1	37.0	84.1	93.7	54.3	79.3
Qwen/Qwen3.6-27B (thinking)	24.4	60.4	26.3	61.8	62.3	83.6	83.1	93.2	-51.0	39.4	-3.1	37.0	84.1	93.7	54.3	79.3
aisingapore/Gemma-SEA-LION-v4.5-E2B-IT	7.1	46.5	-4.5	35.6	37.3	69.3	24.4	60.5	-52.0	38.7	0.4	40.4	5.6	45.2	-10.0	29.8
aisingapore/Gemma-SEA-LION-v4.5-E2B-IT (thinking)	6.0	45.5	-5.3	34.7	39.3	70.6	22.7	59.2	-52.0	38.7	0.9	40.8	5.2	44.8	-9.5	30.4
aisingapore/Qwen-SEA-LION-v4.5-27B-IT	25.6	61.3	27.5	62.7	59.3	82.0	85.8	94.4	-50.0	40.0	-0.9	39.1	85.7	94.3	54.7	79.5
aisingapore/Qwen-SEA-LION-v4.5-27B-IT (thinking)	25.6	61.3	27.2	62.4	59.0	81.8	85.8	94.4	-50.0	40.0	-1.1	38.9	85.2	94.1	53.1	78.6
google/gemma-3-4B-it	21.8	58.5	25.3	61.1	52.3	78.2	73.3	88.9	-46.0	42.5	1.3	41.3	55.1	79.7	57.3	81.0
google/gemma-3-12B-it	21.7	58.4	33.5	66.8	62.3	83.6	85.8	94.4	-53.0	38.1	0.0	40.0	79.7	91.8	58.7	81.7
google/gemma-3-27B-it	**31.6**	**65.5**	**35.7**	**68.2**	**70.0**	**87.3**	**88.4**	**95.5**	-37.0	47.9	0.0	40.0	**89.6**	**95.9**	52.8	78.5
google/gemma-4-26B-A4B-it	9.1	48.2	-7.5	32.5	36.7	68.9	35.1	67.8	-50.0	40.0	0.7	40.6	6.0	45.6	-9.3	30.5
google/gemma-4-26B-A4B-it (thinking)	8.9	48.1	-5.7	34.4	38.7	70.1	36.9	69.0	-52.0	38.7	**1.6**	**41.5**	3.9	43.6	-8.4	31.5
google/gemma-4-31B-it	0.6	40.6	-7.7	32.3	41.7	72.0	41.3	71.8	-39.0	46.7	1.1	41.1	10.7	49.6	-11.1	28.6
google/gemma-4-31B-it (thinking)	0.2	40.2	-7.5	32.5	41.3	71.8	40.4	71.2	-38.0	47.3	1.3	41.3	10.3	49.3	-10.5	29.2
google/gemma-4-E2B-it	6.8	46.3	-4.7	35.4	39.7	70.8	27.1	62.4	-53.0	38.1	-0.2	39.8	6.4	45.9	-10.1	29.6
google/gemma-4-E2B-it (thinking)	6.5	46.0	-4.5	35.6	38.3	69.9	28.0	63.0	-53.0	38.1	0.7	40.6	6.5	46.0	-10.7	29.1
google/gemma-4-E4B-it	6.7	46.2	-2.5	37.6	31.0	65.1	39.6	70.7	-42.0	45.0	-1.8	38.3	-4.0	36.1	-12.7	26.8
google/gemma-4-E4B-it (thinking)	6.1	45.7	-1.8	38.2	32.7	66.2	39.6	70.7	-42.0	45.0	-1.6	38.5	-3.6	36.5	-12.5	27.0
microsoft/Phi-4-mini-instruct	19.0	56.4	5.8	45.4	41.0	71.6	77.8	90.9	-46.0	42.5	-1.8	38.3	36.4	68.7	63.7	84.3
microsoft/Phi-4	23.2	59.6	16.0	54.0	35.0	67.8	82.2	92.9	-56.0	36.1	-3.1	37.0	54.4	79.4	**79.6**	**91.7**
mistralai/Mistral-Small-4-119B-2603	6.2	45.8	14.7	52.9	19.7	56.9	49.3	76.5	-46.0	42.5	-0.2	39.8	3.2	43.0	38.8	70.2
mistralai/Mistral-Small-4-119B-2603 (thinking)	8.2	47.5	12.5	51.2	16.0	54.0	46.7	75.0	**-28.0**	**52.9**	-3.3	36.7	-0.4	39.6	31.6	65.5
openai/gpt-oss-120b (thinking)	23.2	59.6	19.2	56.5	20.7	57.7	75.1	89.7	-44.0	43.7	-1.3	38.7	52.5	78.3	77.5	90.8
openai/gpt-oss-20b (thinking)	23.4	59.7	11.3	50.2	38.0	69.7	81.3	92.5	-56.0	36.1	-1.8	38.3	41.3	71.8	74.8	89.6

Comp = PACUTE Composition; *Hier* = Hierarchical; *LGame* = LangGame; *MDA* = Multi-Digit Addition; *MExt* = PACUTE Morphological Extraction; *MProd* = PACUTE Morphological Production; *Manip* = PACUTE Manipulation; *Syll* = PACUTE Syllabification; Bold = best in column. Cell colour: low → white → high per column.

Table 9: Results for IT models on PACUTE GEN benchmarks. EM = exact match; CM = contains match. All values are percentages.

Model	Comp-Gen		Manip-Gen		MExt-Gen		MProd-Gen		Syll-Gen		Hier-Gen		LGame-Gen		MDA-Gen		CUTE-Gen	
	EM	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*
Qwen/Qwen3-4B	70.2	70.5	47.5	49.9	33.0	48.8	19.3	19.3	1.5	45.0	44.7	56.8	90.6	91.6	99.8	99.8	71.0	72.2
Qwen/Qwen3-4B (thinking)	88.4	88.5	9.6	12.0	37.8	50.2	25.3	25.3	18.5	24.0	53.0	63.8	13.5	14.4	0.8	0.8	7.8	8.3
Qwen/Qwen3-8B	72.9	72.9	51.2	52.9	37.0	51.5	30.0	30.0	7.0	31.5	33.8	46.0	93.6	93.6	99.9	99.9	69.3	69.9
Qwen/Qwen3-8B (thinking)	88.0	88.0	83.1	83.8	42.0	52.0	30.7	31.3	29.5	34.5	51.5	62.3	99.8	99.8	100.0	100.0	88.7	89.0
Qwen/Qwen3-14B	78.9	78.9	59.5	60.8	50.2	55.2	31.3	32.0	28.5	36.0	40.0	55.5	95.1	95.1	98.7	98.7	75.1	75.5
Qwen/Qwen3-14B (thinking)	87.8	87.8	85.2	85.9	51.7	55.2	33.3	33.3	36.0	39.5	52.3	66.2	99.9	99.9	**100.0**	**100.0**	92.5	93.0
Qwen/Qwen3-32B	81.8	81.8	61.9	62.5	47.2	51.5	28.0	28.0	24.5	38.5	43.0	56.7	95.3	95.4	100.0	100.0	78.6	79.1
Qwen/Qwen3-32B (thinking)	87.5	87.5	83.9	84.1	53.0	56.5	30.7	30.7	31.5	42.5	54.2	67.3	98.6	99.9	100.0	100.0	92.6	92.7
Qwen/Qwen3.6-27B	87.6	87.6	75.2	76.5	53.0	59.8	39.3	40.0	44.0	47.0	53.8	69.8	99.5	99.8	99.9	99.9	82.1	85.1
Qwen/Qwen3.6-27B (thinking)	88.9	88.9	92.4	92.6	58.0	60.8	46.7	46.7	37.0	42.5	55.3	67.5	99.8	99.8	100.0	100.0	67.6	96.1
aisingapore/Gemma-SEA-LION-v4.5-E2B-IT	69.5	69.5	55.2	57.2	38.0	53.2	26.0	26.7	15.5	34.0	42.2	57.0	93.3	94.2	99.5	99.5	67.9	70.4
aisingapore/Gemma-SEA-LION-v4.5-E2B-IT (thinking)	68.9	68.9	55.5	57.8	37.8	52.5	24.0	24.7	15.0	35.5	42.0	57.5	94.4	94.5	99.4	99.4	67.1	70.1
aisingapore/Qwen-SEA-LION-v4.5-27B-IT	88.2	88.2	77.9	78.5	55.5	60.0	42.0	42.0	33.5	36.5	52.2	67.5	99.3	99.5	100.0	100.0	83.4	85.8
aisingapore/Qwen-SEA-LION-v4.5-27B-IT (thinking)	88.9	88.9	89.8	89.8	57.8	61.0	44.7	44.7	27.0	29.0	56.7	68.0	97.6	97.6	98.6	98.6	73.6	94.3
google/gemma-3-4B-it	55.1	55.1	44.0	46.1	43.0	53.2	14.7	14.7	15.5	48.5	24.0	37.0	88.9	88.9	77.3	77.4	51.4	57.3
google/gemma-3-12B-it	70.4	70.4	54.9	55.9	48.2	57.0	35.3	35.3	27.5	48.0	36.8	46.8	96.9	96.9	98.5	98.5	72.7	73.7
google/gemma-3-27B-it	75.8	75.8	59.6	60.6	54.0	64.0	44.7	44.7	21.5	30.0	39.0	54.3	96.9	96.9	100.0	100.0	77.2	77.8
google/gemma-4-26B-A4B-it	88.5	88.5	81.2	81.4	79.2	79.8	52.7	52.7	25.5	25.5	53.8	**71.0**	99.7	**100.0**	99.1	99.2	89.8	91.5
google/gemma-4-26B-A4B-it (thinking)	88.7	88.7	81.1	81.4	79.5	80.2	52.7	52.7	25.0	25.0	53.5	69.2	**100.0**	100.0	99.1	99.1	90.8	91.3
google/gemma-4-31B-it	87.8	87.8	87.5	87.6	80.0	80.5	**54.7**	**54.7**	48.5	50.5	53.2	69.7	99.9	99.9	100.0	100.0	66.6	93.9
google/gemma-4-31B-it (thinking)	87.8	87.8	87.4	87.5	**80.2**	**80.8**	54.0	54.0	**49.0**	52.0	53.7	69.3	99.9	99.9	100.0	100.0	66.6	94.1
google/gemma-4-E2B-it	70.9	70.9	53.8	56.0	40.0	55.0	27.3	27.3	14.0	33.5	46.0	59.8	93.4	94.2	99.5	99.5	66.9	70.1
google/gemma-4-E2B-it (thinking)	70.2	70.2	53.2	55.5	40.2	55.2	27.3	27.3	17.0	32.0	45.0	59.2	93.8	93.9	99.4	99.4	67.1	70.4
google/gemma-4-E4B-it	80.7	80.7	59.6	61.1	46.0	51.7	39.3	39.3	33.0	35.5	48.7	64.3	94.5	94.5	99.8	99.8	76.9	78.5
google/gemma-4-E4B-it (thinking)	80.4	80.4	59.9	61.4	45.8	52.0	40.0	40.0	32.0	38.0	48.5	64.7	94.6	94.6	99.8	99.8	77.2	78.3
microsoft/Phi-4-mini-instruct	48.5	48.7	37.5	39.6	25.0	44.2	12.7	14.0	7.5	38.5	18.0	30.8	73.3	73.3	99.9	99.9	42.2	45.7
microsoft/phi-4	86.5	86.5	64.0	64.5	43.2	50.2	31.3	32.7	27.0	36.0	42.0	61.7	95.5	95.5	99.9	99.9	75.9	76.5
mistralai/Mistral-Small-4-119B-2603	80.9	80.9	64.9	66.1	57.8	62.7	40.7	41.3	38.5	50.5	47.8	59.8	96.3	96.3	99.1	99.1	78.1	78.4
mistralai/Mistral-Small-4-119B-2603 (thinking)	81.8	81.8	65.4	66.1	57.0	62.3	42.0	42.7	36.0	**53.0**	46.0	57.8	96.1	96.1	98.0	98.0	79.4	79.8
openai/gpt-oss-120b (thinking)	**91.1**	**91.1**	**94.4**	**94.4**	64.8	67.0	53.3	53.3	45.5	46.0	**59.8**	70.8	99.7	99.7	100.0	100.0	**95.7**	**96.5**
openai/gpt-oss-20b (thinking)	89.1	89.1	82.0	82.2	21.2	28.7	6.7	7.3	9.0	19.0	24.5	28.5	95.2	96.2	100.0	100.0	20.9	23.7

Comp = PACUTE Composition; *Hier* = Hierarchical; *LGame* = LangGame; *MDA* = Multi-Digit Addition; *MExt* = PACUTE Morphological Extraction; *MProd* = PACUTE Morphological Production; *Manip* = PACUTE Manipulation; *Syll* = PACUTE Syllabification; Bold = best in column. Cell colour: low → white → high per column.

Table 10: Results for commercial models on PACUTE GEN benchmarks. EM = exact match; CM = contains match. Commercial APIs do not expose token log-probabilities, so MCQ benchmarks are omitted. All values are percentages.

Model	**Comp-Gen**		**Manip-Gen**		**MExt-Gen**		**MProd-Gen**		**Syll-Gen**		**Hier-Gen**		**LGame-Gen**		**MDA-Gen**		**CUTE-Gen**	
	EM	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*	*EM*	*CM*
claude-haiku-4-5	99.8	**100.0**	82.0	82.0	58.2	61.8	44.0	44.0	69.8	74.2	52.5	71.8	99.7	99.7	**100.0**	**100.0**	88.4	89.1
claude-sonnet-4-6	90.4	90.4	94.1	94.2	74.0	76.2	54.0	54.0	58.0	60.5	57.2	68.0	99.9	99.9	**100.0**	**100.0**	91.9	93.9
claude-opus-4-6	**100.0**	**100.0**	98.5	98.8	77.5	81.2	64.7	66.0	75.6	78.6	55.7	76.2	99.9	99.9	**100.0**	**100.0**	96.6	97.9
deepseek/deepseek-r1-0528 (thinking)	89.5	89.5	96.9	96.9	75.5	76.0	62.0	62.0	55.0	55.0	60.3	72.7	**100.0**	**100.0**	**100.0**	**100.0**	79.3	80.8
deepseek/deepseek-v3.2-exp (thinking)	88.9	88.9	84.1	84.4	74.5	77.0	56.0	56.0	41.0	42.5	53.5	67.3	99.7	99.7	99.9	99.9	90.1	90.4
google/gemini-3.1-flash-lite	86.7	86.7	83.1	83.8	79.5	80.5	63.3	63.3	36.5	38.0	52.5	66.2	99.4	99.4	**100.0**	**100.0**	79.4	90.2
google/gemini-3.5-flash (thinking)	90.4	90.4	98.2	98.2	**89.5**	**90.5**	**90.0**	**90.0**	82.5	82.5	47.2	62.2	99.3	99.3	99.9	99.9	45.4	97.9
gpt-5-mini	90.4	90.4	99.5	99.5	74.5	75.2	60.7	60.7	44.5	46.0	55.2	67.7	**100.0**	**100.0**	**100.0**	**100.0**	89.6	98.7
gpt-5.4-nano	82.2	82.2	67.6	68.5	44.8	53.2	36.0	36.0	32.5	35.0	42.8	52.5	94.9	94.9	99.4	99.4	74.4	75.1
gpt-5.4-mini	86.0	86.2	77.6	78.0	60.8	65.0	52.7	52.7	41.0	41.0	56.0	69.7	98.5	98.5	99.7	99.7	87.4	87.6
gpt-5.5	90.9	90.9	**100.0**	**100.0**	77.0	77.8	82.0	82.0	**83.0**	**83.0**	**64.7**	**77.5**	**100.0**	**100.0**	**100.0**	**100.0**	**99.3**	**99.3**
moonshotai/kimi-k2-thinking (thinking)	88.9	88.9	94.8	94.8	73.2	74.5	60.0	60.0	54.0	58.5	56.5	67.2	99.4	99.6	99.9	99.9	89.1	94.3

Comp = PACUTE Composition; *Hier* = Hierarchical; *LGame* = LangGame; *MDA* = Multi-Digit Addition; *MExt* = PACUTE Morphological Extraction; *MProd* = PACUTE Morphological Production; *Manip* = PACUTE Manipulation; *Syll* = PACUTE Syllabification; Bold = best in column. Cell colour: low → white → high per column.

E Continued Pre-training Results

Results from continued pre-training of Gemma-2-2B on SEA-PILE v2 Filipino corpus (∼7.4GB) across three tokenization regimes: Vanilla (standard BPE), StochasTok (∼10% stochastic token expansion), and Patok (morphology-aware expand/contract). MCQ chance = 25% (4-way). All CPT checkpoints lose math capability entirely (catastrophic forgetting), confirming the multi-digit addition benchmark as a reliable domain-adaptation health check.

Table 11: MCQ benchmark accuracy for Gemma-2-2B and its three CPT regimes on SEA-PILE v2 (5,000 steps). All values are raw 4-way accuracy percentages (chance = 25%). Bold = best per column. CPT regimes broadly underperform the un-CPT'd base on character-level tasks and the LangGame/MDA controls, with MDA in particular dropping from 68.7% to 14–18% (catastrophic forgetting on arithmetic).

Regime	**Comp**	**Manip**	**MExt**	**MProd**	**Syll**	**PACUTE Avg.**	**Hier**	**LGame**	**MDA**
Base (no CPT)	**36.6**	27.3	**66.5**	82.7	**32.0**	**49.0**	23.5	**41.0**	**68.7**
Vanilla BPE	25.8	34.9	57.5	**85.3**	21.0	44.9	22.5	28.0	17.4
StochasTok	20.1	32.5	50.7	78.0	23.5	41.0	21.2	28.2	14.8
Patok	27.6	**35.2**	54.3	78.0	19.5	42.9	**24.7**	27.1	17.9

Comp/Manip/MExt/MProd/Syll = PACUTE main suite MCQ tasks; *PACUTE Avg.* is the macro-average of these five.
Hier = Hierarchical diagnostic; *LGame* = LangGame (language-agnostic control); *MDA* = Multi-Digit Addition (catastrophic-forgetting probe).

F Error Analysis: Annotated Examples

Tables 12–15 provide representative model outputs for each of the four error types described in §7. All examples are drawn from GEN inference outputs of instruction-tuned models. Task inputs are quoted verbatim; model outputs are lightly formatted for readability (angle-bracket tags are from the model's structured output format).

Table 12: Type I: Instruction following failure. Model produces a malformed output—here, a character-insertion task yields garbled output with spurious spaces rather than the target string.

Input	**Model output (Qwen3-8B)**
Put a character "b" after every character "o" in the word "tungkol".	<reflection>I identified the operation as inserting "b" after each "o".</reflection> <answer>tu ngbk b l</answer>
Gold answer: *tungkobl*	(exact match: ✗; contains match: ✗)

Table 13: Type II: Linguistic error. The model correctly identifies that *-in-* is an infix and that it should be placed inside the root, but inserts it at the wrong position (*tak**in**bo* vs. the correct ***tin**akbo*, where the infix follows the initial consonant).

Input	**Model output (Gemma-3-27B-IT)**
Inflect the word "takbo" to use the infix "-in-".	<reflection>Infixation inserts an affix within the root word; *-in-* results in *takinbo*.</reflection> <answer>takinbo</answer>
Gold answer: *tinakbo*	(exact match: ✗; contains match: ✗)

Table 14: Type III: Reasoning inconsistency. The model's chain-of-thought correctly derives *kinain* ("k + in + ain") but then dismisses this intermediate result and outputs *kalinin* as the final answer.

Input	**Model output (Qwen3-8B-Thinking)**
Inflect the word "kain" to use the infix "-in-".	<reflection>... inserting *-in-* between 'k' and 'a': "k" + "in" + "ain"? that would be *kinain*? wait, but *kinain* is the past tense ... maybe the correct answer is *kalinin*.</reflection> <answer>kalinin</answer>
Gold answer: *kinain*	(exact match: ✗; contains match: ✗)

Table 15: Type IV: Systematic penultimate-stress bias. The model explicitly asserts that Filipino stress defaults to the penultimate syllable and selects *ka* accordingly. In the given sentence, *kaya* means "which is why" (*kayà*), which carries final stress on *ya*.

Input	**Model output (Gemma-3-27B-IT)**
Based on the sentence "Ito ang unang beses ko, kaya ako kabada.", which syllable of the word "kaya" has the stress?	<reflection>the sentence indicates "kaya" means "can" or "able to", implying a verb usage. in filipino, verbs generally have stress on the penultimate (second to last) syllable.</reflection><answer>ka</answer>
Gold answer: *ya*	(exact match: ✗; contains match: ✗)