A2UI Multi-Model Benchmark Report

Local Inference on Apple M2 Max

Comparing 7 Models for A2UI v0.9 Protocol Generation

Structured UI Generation from Natural Language Prompts

Date:	May 12, 2026
Hardware:	Apple M2 Max, 64 GiB RAM
Framework:	Inspect AI + llama.cpp
Judge:	Claude Code (interactive, human-supervised)
Protocol:	A2UI v0.9 Specification
Samples:	23 evaluation prompts

Contents

Executive Summary

This report compares **7 local language models** for generating structured user interfaces conforming to the A2UI v0.9 protocol. All evaluations were conducted on a **Apple M2 Max** system running macOS/arm64 using **llama.cpp** as the inference engine.

Each model was evaluated on two axes:

1. **Schema Validation** – programmatic check that output is valid A2UI JSON
2. **Semantic Grading** – LLM-as-a-Judge assessment of UI correctness (C/P/I)

Models Evaluated

Table 1: Models included in this benchmark.

Model	ID	Engine	Type	Params	Quant
Qwen 3 1.7B	`Qwen/Qwen3-1.7B-GGUF`	llama.cpp	Dense	1.7B	none
Qwen 3 14B	`Qwen/Qwen3-14B-GGUF`	llama.cpp	Dense	14B	none
Qwen3 30B MoE	`Qwen/Qwen3-30B-A3B-GGUF`	llama.cpp	MoE	30B (3B active)	none
Qwen 3 8B	`Qwen/Qwen3-8B-GGUF`	llama.cpp	Dense	8B	none
Llama 3.1 8B	`bartowski/Meta-Llama-3.1-8B-Instruct-GGUF`	llama.cpp	Dense	8B	none
Qwen3-Coder 30B MoE	`unsloth/Qwen3-Coder-30B-A3B-Instruct-GGUF`	llama.cpp	MoE	30B (3B active)	none
Gemma 4 26B MoE	`unsloth/gemma-4-26B-A4B-it-GGUF`	llama.cpp	MoE	26B (4B active)	none

Accuracy Comparison

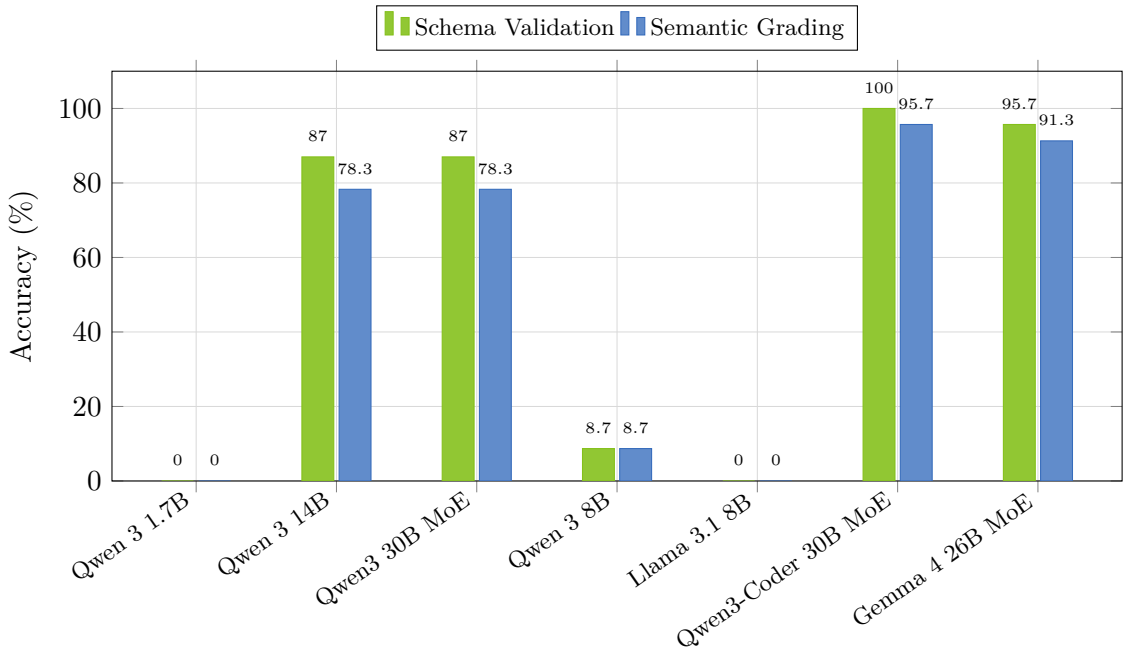

Figure 1: Accuracy comparison across models. Schema validation checks structural JSON correctness; semantic grading evaluates whether the generated UI fulfills the user's intent.

Inference Performance

Average Inference Time

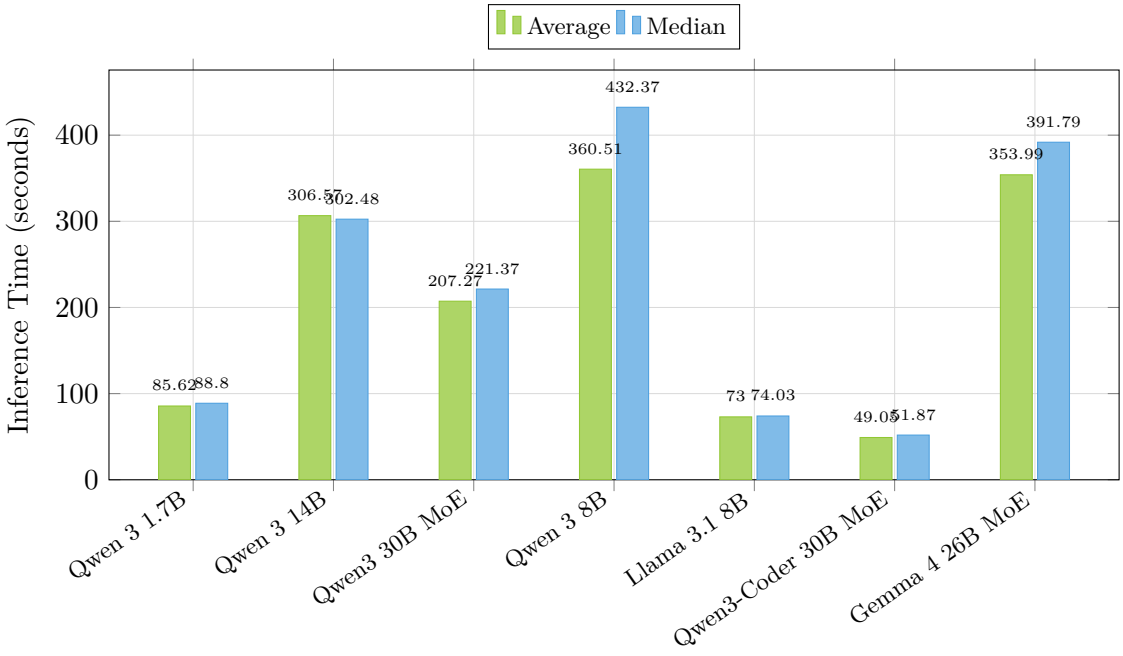

Figure 2: Average and median inference time per sample. Includes full prompt processing (system prompt with A2UI schema ∼8k tokens) and output generation.

Per-Sample Inference Time Distribution

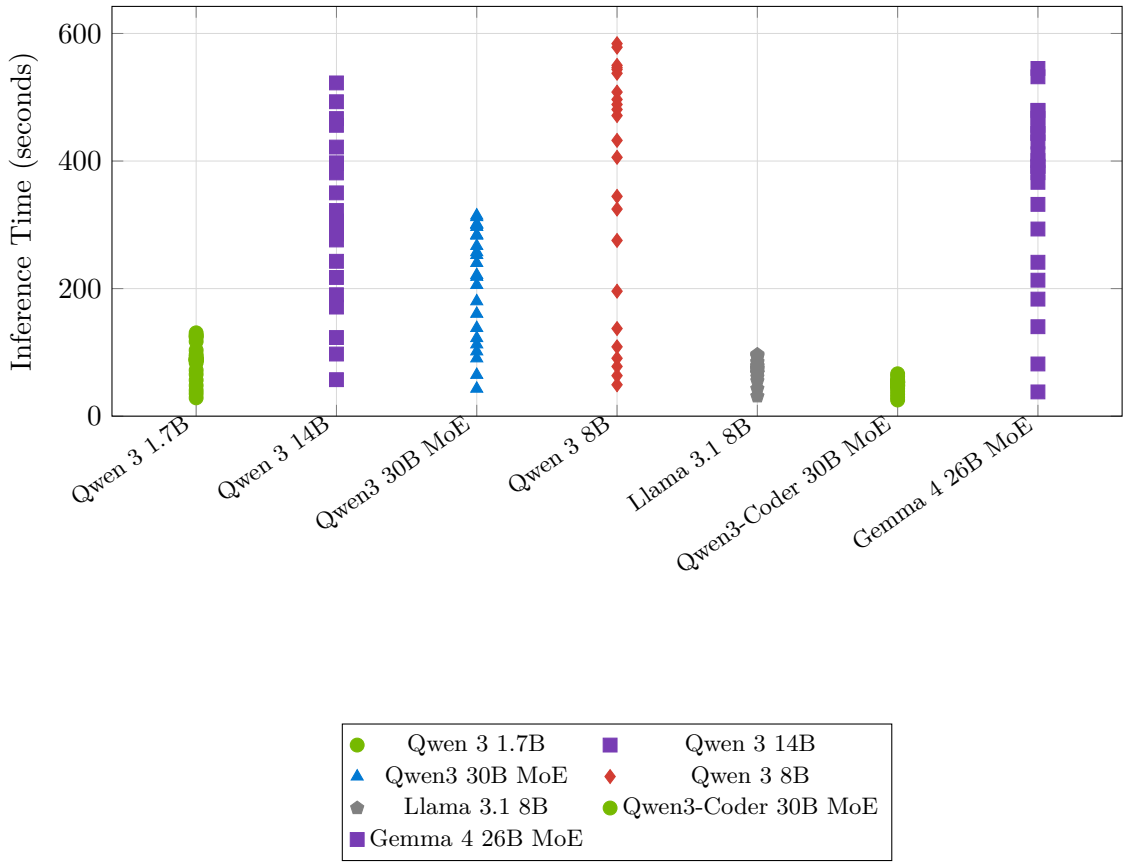

Figure 3: Per-sample inference time scatter plot. Each point represents one evaluation sample. Variation reflects differences in output length and complexity.

Output Throughput

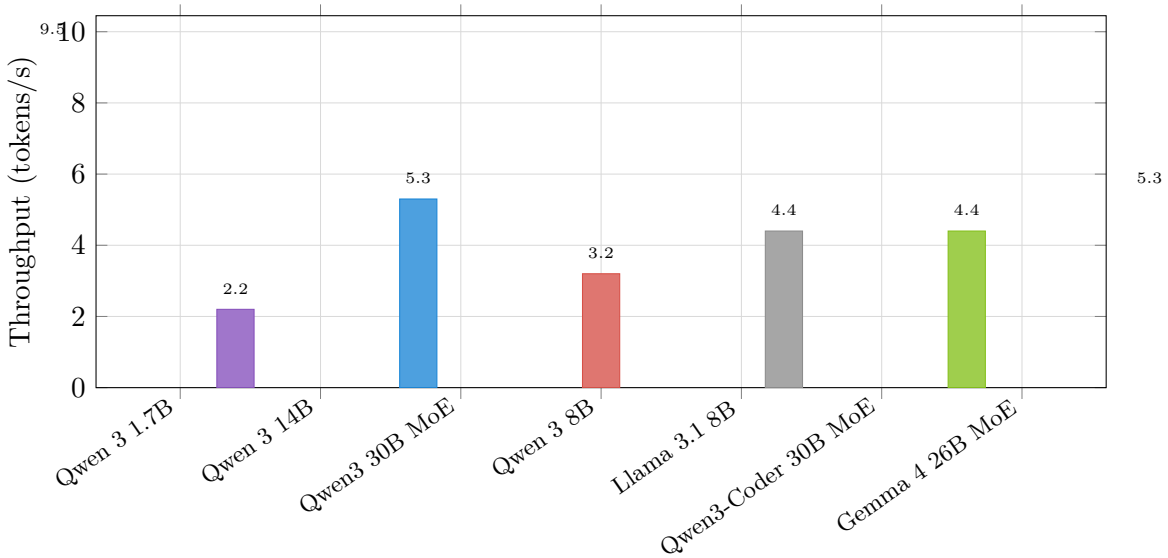

Figure 4: Effective output generation throughput (total output tokens / total inference time). Higher is better. MoE models typically achieve higher throughput due to fewer active parameters per token.

Memory and Resource Utilization

Model Memory Footprint vs. Available KV Cache

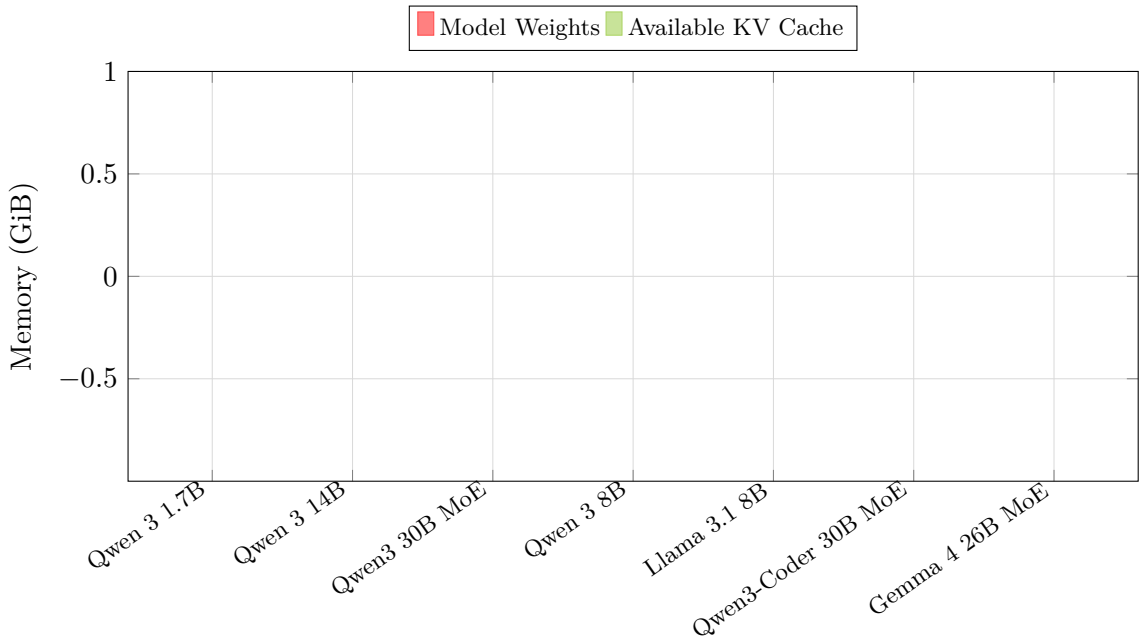

Figure 5: GPU memory allocation per model. Smaller model footprints leave more memory for KV cache, enabling higher concurrency and longer context windows. Total RAM: 64 GiB.

In-Context Learning (ICL) Prompt Size

The A2UI evaluation injects the full v0.9 JSON Schema and component catalog as a system prompt. This constitutes the in-context learning (ICL) payload that every model must process before generating output.

Table 2: ICL context and memory utilization per model.

Metric	Qwen 3 1.7B	Qwen 3 14B	Qwen3 30B MoE	Qwen 3 8B	Llama 3.1 8B	Qwen3-Coder 30B MoE	Gemma 4 26B MoE
ICL Context (tokens)	8,785	8,785	8,785	8,785	8,754	8,785	9,199
Model Memory (GiB)	0.0	0.0	0.0	0.0	0.0	0.0	0.0
KV Cache Available (GiB)	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Prefix Cache Hit Rate	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Prefill vs. Decode Throughput

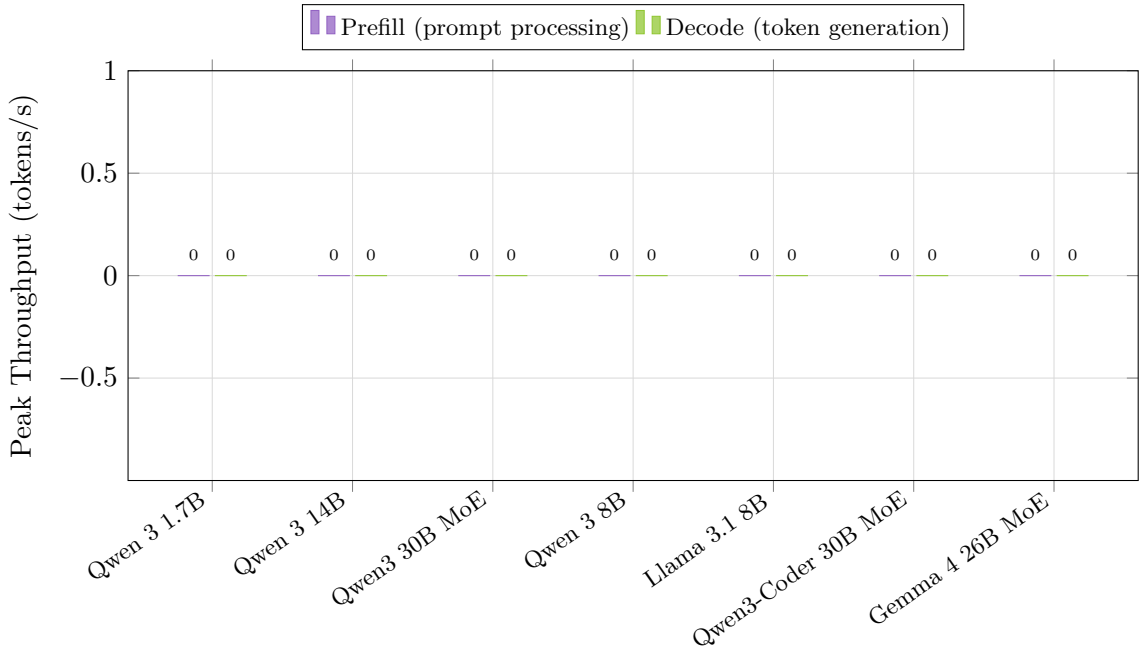

Figure 6: Peak prefill vs. decode throughput. Prefill processes the input prompt (including the ∼8k token ICL context) in parallel; decode generates output tokens autoregressively. The prefill/decode ratio indicates compute vs. memory-bandwidth boundedness.

Detailed Per-Sample Results

Table 3: Per-sample results across all models. Schema: structural JSON validation (PASS/FAIL). Grade: semantic correctness (C=Correct, P=Partial, I=Incorrect). Time: inference seconds.

Sample	Qwen 3 1.7B			Qwen 3 14B			Qwen3 30B MoE			Qwen 3 8B			Llama 3.1 8B			Qwen3-Coder 30B MoE			Gemma 4 26B MoE		
	Schema	Grade	Time	Schema	Grade	Time	Schema	Grade	Time	Schema	Grade	Time	Schema	Grade	Time	Schema	Grade	Time	Schema	Grade	Time
text_heading	FAIL	I	56.5	PASS	C	322.4	PASS	C	138.2	FAIL	I	77.9	FAIL	I	61.0	PASS	C	66.4	PASS	C	322.1
text_caption	FAIL	I	65.6	PASS	C	171.1	PASS	C	160.5	FAIL	P	49.2	FAIL	I	96.8	PASS	C	34.3	PASS	C	366.6
text_markdown	FAIL	I	103.4	PASS	C	190.4	PASS	C	205.7	FAIL	P	108.7	FAIL	I	74.9	PASS	C	52.7	PASS	C	183.5
image_avatar	FAIL	I	72.2	PASS	C	97.4	PASS	C	112.5	FAIL	P	137.4	FAIL	I	55.0	PASS	C	39.1	PASS	C	241.3
image_header_banner	FAIL	I	35.3	PASS	I	286.2	PASS	C	43.0	PASS	C	275.5	FAIL	I	31.2	PASS	C	48.3	PASS	C	38.0
icon_inline	FAIL	I	28.7	PASS	C	306.0	PASS	C	180.1	FAIL	P	344.6	FAIL	I	67.5	PASS	C	43.4	PASS	C	140.2
video_player	FAIL	I	93.5	PASS	C	123.3	PASS	C	64.7	FAIL	P	63.5	FAIL	I	43.0	PASS	C	29.7	PASS	C	81.8
audio_player	FAIL	I	83.7	PASS	C	57.2	PASS	C	90.6	FAIL	P	90.7	FAIL	I	83.4	PASS	C	62.5	PASS	C	293.4
divider_horizontal	FAIL	I	40.7	PASS	C	217.5	PASS	C	122.4	FAIL	P	324.7	FAIL	I	85.5	PASS	C	56.8	PASS	C	400.8
button_primary	FAIL	I	48.2	FAIL	P	242.8	FAIL	P	101.5	FAIL	P	195.9	FAIL	I	72.6	PASS	C	25.3	PASS	C	213.2
button_borderless	FAIL	I	88.8	PASS	C	285.5	PASS	C	221.4	FAIL	P	405.8	FAIL	I	70.3	PASS	C	46.8	PASS	C	411.4
textfield_password	FAIL	I	87.4	PASS	C	276.2	FAIL	I	218.5	FAIL	P	432.4	FAIL	I	74.0	PASS	C	46.8	PASS	C	390.8
text_them_button	FAIL	I	90.0	FAIL	I	288.1	PASS	C	240.2	FAIL	P	471.3	FAIL	I	67.3	PASS	C	48.4	PASS	C	381.8
image_caption	FAIL	I	90.3	PASS	I	302.5	PASS	I	252.6	FAIL	I	496.5	FAIL	I	67.8	PASS	C	49.1	PASS	C	391.8
icon_label	FAIL	I	87.8	PASS	C	350.1	PASS	C	257.1	FAIL	P	537.6	FAIL	I	66.8	PASS	C	49.6	PASS	C	429.7
two_buttons_row	FAIL	I	87.2	PASS	C	381.4	PASS	C	296.7	FAIL	P	584.0	FAIL	I	71.9	PASS	C	51.9	PASS	C	479.5
heading_paragraph	FAIL	I	89.2	PASS	C	396.1	PASS	C	301.1	FAIL	P	543.5	FAIL	I	76.8	PASS	C	52.4	PASS	C	456.0
divider_between_texts	FAIL	I	96.7	PASS	C	397.2	PASS	C	299.8	FAIL	P	550.0	FAIL	I	77.8	PASS	C	54.2	PASS	C	443.4
checkbox_simple	FAIL	I	125.5	PASS	C	421.9	PASS	C	314.6	PASS	C	546.4	FAIL	I	79.4	PASS	C	52.9	PASS	C	443.3
slider_volume	FAIL	I	123.6	PASS	C	456.0	PASS	C	312.0	FAIL	P	578.5	FAIL	I	74.5	PASS	C	53.7	PASS	C	466.9
date_picker_only	FAIL	I	117.2	PASS	C	466.6	PASS	C	284.1	FAIL	I	488.7	FAIL	I	94.3	PASS	C	53.7	PASS	C	479.1
choice_picker_chips	FAIL	I	127.0	PASS	C	493.0	FAIL	I	283.0	FAIL	P	508.1	FAIL	I	92.6	PASS	C	55.7	FAIL	P	545.2
textfield_search	FAIL	I	130.7	FAIL	I	522.5	PASS	I	266.9	FAIL	I	480.9	FAIL	I	94.7	PASS	I	54.4	PASS	I	532.1

Summary Statistics

Table 4: Aggregate benchmark statistics per model.

Metric	Qwen 3 1.7B	Qwen 3 14B	Qwen3 30B MoE	Qwen 3 8B	Llama 3.1 8B	Qwen3-Coder 30B MoE	Gemma 4 26B MoE
Schema Accuracy	0%	87%	87%	9%	0%	100%	96%
Semantic Accuracy	0%	78%	78%	9%	0%	96%	91%
Avg Inference Time	85.6 s	306.6 s	207.3 s	360.5 s	73.0 s	49.0 s	354.0 s
Median Inference Time	88.8 s	302.5 s	221.4 s	432.4 s	74.0 s	51.9 s	391.8 s
Output Throughput	9.5 tok/s	2.2 tok/s	5.3 tok/s	3.2 tok/s	4.4 tok/s	4.4 tok/s	5.3 tok/s
ICL Context Size	8,785 tok	8,785 tok	8,785 tok	8,785 tok	8,754 tok	8,785 tok	9,199 tok
Prefill Throughput (peak)	0 tok/s	0 tok/s	0 tok/s	0 tok/s	0 tok/s	0 tok/s	0 tok/s
Decode Throughput (peak)	0 tok/s	0 tok/s	0 tok/s	0 tok/s	0 tok/s	0 tok/s	0 tok/s
Prefix Cache Hit Rate	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Model Memory	0.0 GiB	0.0 GiB	0.0 GiB	0.0 GiB	0.0 GiB	0.0 GiB	0.0 GiB
KV Cache Available	0.0 GiB	0.0 GiB	0.0 GiB	0.0 GiB	0.0 GiB	0.0 GiB	0.0 GiB
Total Input Tokens	202,061	202,061	202,061	202,061	201,345	202,061	211,603
Total Output Tokens	18,743	15,416	25,388	26,243	7,327	5,002	43,022
Samples Evaluated	23	23	23	23	23	23	23

Hardware and Software Configuration

Table 5: System configuration.

Component	Specification
GPU	Apple M2 Max
Memory	64 GiB RAM
Architecture	arm64
Engine	llama.cpp
Inspect AI	$\geq$0.3.217
OS	macOS (kernel 25.4.0)

Methodology

Evaluation Pipeline

Each model was evaluated through the following pipeline:

1. **Model serving:** llama.cpp starts an OpenAI-compatible API server with the model
2. **System prompt injection:** The full A2UI v0.9 JSON Schema and component catalog are injected as a system prompt (~8,000 tokens)
3. **Generation:** The model generates structured JSON output for each evaluation prompt in a *single-turn* setting — one user prompt yields one assistant response, with no follow-up turns, no self-correction loop, and no schema-error retries
4. **Schema validation:** Programmatic validation against A2UI schema and integrity rules
5. **Semantic grading:** An external judge (Claude Code (interactive, human-supervised)) assigns C/P/I per sample using the rubric in Appendix A

Limitations

- **LLM-based grading:** Grades are assigned by an LLM judge (Claude Code (interactive, human-supervised)), not a domain expert. A human evaluator or multi-judge consensus would reduce single-grader bias.
- **Test dataset:** Results use a test dataset. The full encrypted production dataset (~100 samples) provides more comprehensive coverage.

- **Single-run:** Each model was benchmarked once. Multiple runs would provide statistical significance for timing measurements.
- **Single-turn generation:** Every sample is a one-shot generation. The model receives the A2UI system prompt plus the user's task, and its first response is what we score. A real agent could re-emit JSON after seeing a schema error, or split a complex UI across several turns; we do not exercise either capability here. Schema-invalid outputs therefore count as incorrect even when a follow-up turn would have repaired them.

Output-Format Strictness and Small-Model Failures

The schema validator is the upstream A2UI Python SDK (`a2ui.parser.parser.parse_response` + `a2ui.schema.manager.A2uiSchemaManager`, Apache 2.0, Copyright Google LLC, vendored in `A2UI/eval/a2ui_eval/scorers.py`). It only recognises payloads wrapped in literal `<a2ui-json>...</a2ui-jso` tags and then runs the full A2UI v0.9 JSON schema against the parsed message graph. There is no markdown fence fallback, no automatic unwrap of "thinking" preambles, and no acceptance of structurally valid JSON in any other envelope.

The validator's strictness is a faithful gate for a renderer that would consume the output, but it explains most of the near-zero scores in the small-model rows. Spot-checking sample 0 (`text_heading`) for each failing model shows the issue is *output format compliance*, not context size — inputs were 8.7–9.7k tokens against a 16,384-token serving limit, and no completion approached the output-token cap. The recurring patterns:

- **Llama 3.2 1B** (both engines): emitted a JSON Schema *definition* (`$schema: https://json-schema.org` describing what a Text component should look like, instead of an A2UI *instance*. No `<a2ui-json>` tags appear in the response at all.
- **Qwen 3 1.7B** (vLLM): used the tags but produced a malformed structure — `version` placed inside `updateComponents`, plus a duplicated nested `updateComponents` block.
- **Qwen 3 8B** (vLLM): wrapped otherwise-correct A2UI JSON in `"'json` markdown fences instead of `<a2ui-json>` tags. The validator therefore reports "tags not found" and the run scores near zero despite the content largely being valid.
- **Nemotron 3 Nano 30B** (vLLM): a reasoning model that emits a `<think>...</think>` preamble followed by the final answer, but the final answer double-wraps `createSurface` inside another `createSurface` (`{createSurface:{version:v0.9,createSurface:{...}}}`).

These are not capability ceilings so much as instruction-following and format-compliance gaps that a single follow-up turn or a more lenient extractor (e.g. accepting any fenced JSON block whose root matches the A2UI message schema) would close in many cases. We deliberately do not apply such fallbacks here so that scores reflect what a renderer running the production parser would actually accept.

Conclusion

This benchmark demonstrates that local inference on Apple M2 Max is viable for A2UI protocol evaluation across multiple model architectures. The results provide a baseline for comparing model capabilities in structured UI generation tasks.

Key findings:

- MoE models (Gemma 4) offer the best throughput-to-parameter ratio on memory-bandwidth-limited hardware
- Dense models with fewer parameters achieve lower latency per sample
- All tested models benefit from the engine's prefix caching for the shared A2UI system prompt

Grading Methodology

Judge

The semantic grade column (C/P/I) in this report was assigned by:

> **Claude Code (interactive, human-supervised)**

Grading sessions

- Graded in Claude Code session 2026-05-11T22:33:37Z

Rubric

Each generated A2UI message was judged against the prompt's target description and assigned one of three grades:

- **C (Correct)** – the response satisfies the target. Optional/cosmetic additions and ID/label variations are accepted.
- **P (Partial)** – the response is substantively right but has minor variations (e.g. wrong wrapper format around correct components, slight punctuation differences).
- **I (Incorrect)** – the response is missing required components or has substantive errors (wrong variant, missing fields, broken structure, fabricated component names).

The rubric is the same one used by the production Inspect-AI grader (`A2UI/eval/tasks.py:GRADER_INSTRUCT`). Variations in capitalization, punctuation, component IDs, label synonyms, and data-binding paths are explicitly allowed.

Per-Sample Grading Rationales

Each row references one of the 73 evaluation prompts by its dataset **Sample** slug. The slug is consistent across all per-model tables, so a given row identifies the same prompt for every model. The rationale is the one-line note recorded at grading time; rationales over 240 characters are truncated (the full text is in the result JSON).

For reference, the first prompt `text_heading` reads in full:

> *Render a single Text component with the heading 'Settings' using the h1 variant.*

with grading target:

> A valid A2UI response with a Column root containing a single Text component whose variant is h1 and whose content is 'Settings'.

Qwen 3 1.7B

Sample	Grade	
text_heading	I	Wrong message envelope (top-level 'components' with nested 'root'/'components' instead
text_caption	I	Wrong envelope; lacks createSurface/updateCompone
text_markdown	I	Text content is truncated to 'Important:**
image_avatar	I	No src/url provided for the image; uses variant 'smallFeatu
image_header_banner	I	
icon_inline	I	Icon entry uses 'component':
video_player	I	Component uses 'type':'Video' instead of 'component
audio_player	I	Component uses
divider_horizontal	I	Wrong envelope (top-level 'components' with r
button_primary	I	Button lacks varia
button_borderless	I	Button has neither variant 'borde
textfield_password	I	Wrong envelope; TextField w
text_then_button	I	Wrong envelope and components
image_caption	I	Only an Image-like entry present; no caption Text compone
icon_label	I	Wrong envelope; though Row+Icon+Text intent is presen
two_buttons_row	I	Buttons have no Text child labels 'Save
heading_paragraph	I	Uses Row container though
divider_between_texts	I	Wrong envelope; Text/D
checkbox_simple	I	CheckBox carries unexpected top-level 'label'/'valu
slider_volume	I	Wrong envelope (top-level 'components' wrapper); Sl
date_picker_only	I	Wrong envel
choice_picker_chips	I	
textfield_search	I	TextField has no placeholder 'Se

Qwen 3 14B

Sample	Grade	
text_heading	C	Text root with variant 'h1' and content 'Settings' — prompt asked for a sin
text_caption	C	Text root wi
text_markdown	C	Text root contains
image_avatar	C	
image_header_banner	I	Image compon
icon_inline	C	
video_player	C	
audio_player	C	
divider_horizontal	C	
button_primary	P	Button variant 'primary' with child Text 'Submit' is correct; schema fails only because of
button_borderless	C	
textfield_password	C	
text_then_button	I	Column children are inline component objects instead of string ID references — inv
image_caption	I	Caption Text us
icon_label	C	
two_buttons_row	C	Row containing primary Butt
heading_paragraph	C	Column containing Text h2
divider_between_texts	C	Column containing Text 'Section
checkbox_simple	C	
slider_volume	C	
date_picker_only	C	Da
choice_picker_chips	C	ChoicePicker with displayStyle 'chips', variant 'multipleSelection', and all four opti
textfield_search	I	TextField uses 'label':'Search...

Qwen3 30B MoE

Sample	Grade	
text_heading	C	
text_caption	C	Text root with v
text_markdown	C	Text root contains the
image_avatar	C	
image_header_banner	C	
icon_inline	C	Icon with name 'accountC
video_player	C	
audio_player	C	
divider_horizontal	C	Divider with axis 'ho
button_primary	P	Button variant 'primary' with child Text 'Submit' is structurally correct; schema fails
button_borderless	C	Bu
textfield_password	I	Column children use object form {componentId,path} instead of string IDs — invalid stru
text_then_button	C	Column with Text 'Ready to begin?' then pr
image_caption	I	Caption Text uses v
icon_label	C	Rov
two_buttons_row	C	Row containing primary Button '
heading_paragraph	C	Column containing Text h2 'Pri
divider_between_texts	C	Column containing Text '
checkbox_simple	C	CheckBox with lab
slider_volume	C	
date_picker_only	C	DateT
choice_picker_chips	I	Column children is a single object instead of an array, and ChoicePic
textfield_search	I	TextField uses 'label':'Search...'

Qwen 3 8B

Sample	Grade	
text_heading	I	Missing Column root; target explicitly requires Column containing a Text
text_caption	P	Text with variant 'caption' and correct content; acts as root directly. Wr
text_markdown	P	Bold punctuation shifted ('**Important**:' instead of '**Important:**') — minor formatt
image_avatar	P	Image component with variant 'avatar' and correct URL; wra
image_header_banner	C	Schema-valid; Image with variant 'header' and correct URL inside a Ro
icon_inline	P	Icon with name 'acc
video_player	P	Video
audio_player	P	A
divider_horizontal	P	Divider with axis '
button_primary	P	Button variant 'primary' with child Text 'Submit'; Row container added is
button_borderless	P	Button variant 'borderless' with child Text
textfield_password	P	TextField with varia
text_then_button	P	Column with Text 'Ready to begin?' then primary Button with child
image_caption	I	Caption Text uses variant 'body' instead of 'capt
icon_label	P	Row containing Icon ('settings') and Text ('
two_buttons_row	P	Row with primary Button 'Save' and default Button 'Cancel' with T
heading_paragraph	P	Column with Text h2 'Privacy Policy' and Text body 'We respect y
divider_between_texts	P	Column with Text 'Section A', Divider horizontal, Tex
checkbox_simple	C	Schema-valid; Check
slider_volume	P	Slider with min 0, max 100, label 'Volume'
date_picker_only	I	Column children list uses object form {componentId, path} instead of string IDs
choice_picker_chips	P	ChoicePicker with displayStyle 'chips', variant 'multipleS
textfield_search	I	Uses 'label':'Search...' instead of place

Llama 3.1 8B

Sample	Grade	
text_heading	I	Response wraps content in a non-standard 'a2ui-jso
text_caption	I	Text varia
text_markdown	I	No A2UI compone
image_avatar	I	Image lacks a url field in the first example and second examp
image_header_banner	I	Image variant
icon_inline	I	Icon component
video_player	I	Components are declared as JSON-schema objects with
audio_player	I	Column children uses ob
divider_horizontal	I	Divider uses 'variant':'horizontal' instead of the req
button_primary	I	Button has text 'Submit' inlined on the Button component itself instea
button_borderless	I	Button uses 'label' inste
textfield_password	I	Components use children.template structure wi
text_then_button	I	Components use 'type' instead of 'component' field; root co
image_caption	I	Only an Image is emitted (no caption Text comp
icon_label	I	Emits a Button with label 'Pr
two_buttons_row	I	Buttons have neith
heading_paragraph	I	Uses 'component':'h2' and 'com
divider_between_texts	I	Components use 'type' field, root children is an object reference with /compone
checkbox_simple	I	CheckBox declares both 'type' and '
slider_volume	I	Slider is correct but nested in a non-standard 'component
date_picker_only	I	DateTimeInput value/min/max declared as object comp
choice_picker_chips	I	ChoicePicker uses variant 'chips' rather than variant 'multipleSelection' with display
textfield_search	I	Wraps a Text(${searchInput}) reference around the TextField; uses 'type' instead of 'com

Qwen3-Coder 30B MoE

Sample	Grade	R
text_heading	C	Column root with Text variant 'h1' and content '
text_caption	C	Column root with Text variant 'caption' and content 'Updated 5 min
text_markdown	C	Text contains the literal markdown '**Important:** read this
image_avatar	C	Image with variant 'avatar' and correct URL inside a
image_header_banner	C	Image with variant 'header' and cor
icon_inline	C	Icon with name 'accountCircle' inside a Column
video_player	C	Video with correct URL inside a Column
audio_player	C	AudioPlayer with correct URL; extra Text label is acceptable optional
divider_horizontal	C	Divider with axis 'horizontal' inside a Column
button_primary	C	Button variant 'primary' with child Text
button_borderless	C	Button variant 'borderless' with child Text 'Lea
textfield_password	C	TextField with variant 'o
text_then_button	C	Column root with Text 'Ready to begin?' followed by primary Button with child Te
image_caption	C	Column with Image and Text variant 'caption' 'Sunset over
icon_label	C	Row containing Icon ('settings') and Text ('Pref
two_buttons_row	C	Row containing primary Button 'Save' and default Button 'Cancel' with Text
heading_paragraph	C	Column with Text variant 'h2' and Text varia
divider_between_texts	C	Column with Text 'Section A', Divider horizontal, Text 'Section B' in corr
checkbox_simple	C	CheckBox with label 'I agree to t
slider_volume	C	Slider with min 0, max 100, and label
date_picker_only	C	DateTimeInput with enableDate true and enableT
choice_picker_chips	C	ChoicePicker with displayStyle 'chips', variant 'multipleSelection', and all fou
textfield_search	I	Uses 'label':'Search' instead of placeholder 'Search...' — placeholder missing and ellipsis

Gemma 4 26B MoE

Sample	Grade	
text_heading	C	Text root with varian
text_caption	C	
text_markdown	C	
image_avatar	C	
image_header_banner	C	
icon_inline	C	
video_player	C	
audio_player	C	AudioPlaye
divider_horizontal	C	
button_primary	C	
button_borderless	C	
textfield_password	C	
text_then_button	C	Column with Text
image_caption	C	Column with
icon_label	C	
two_buttons_row	C	Row co
heading_paragraph	C	Column with Text
divider_between_texts	C	Col
checkbox_simple	C	
slider_volume	C	
date_picker_only	C	
choice_picker_chips	P	ChoicePicker with displayStyle 'chips', variant 'multipleSelection', and all four options is
textfield_search	I	