

SK Telecom releases A.X K2, a 688bn parameter model from Korea's sovereign AI project

Source copy kept by wheresthe.ai for the story "SK Telecom releases A.X K2, a 688bn parameter model from Korea's sovereign AI project" (<https://wheresthe.ai/d/kr-skt-ax-k2-sovereign-model-2026-07/>).

Source: <https://huggingface.co/skt/A.X-K2>

Text saved 18 Oct 2026 (the page as published).

SHA-256 of the saved text: 893f43b462746fa59d19a47c27f9a396858be464607cc13919e4dae2408d310e

```
","cls_token":"<|cls|>","eos_token":"<|im_end|>","mask_token":"<|mask|>","pad_token":"<|pad|>","sep_token":"<|sep|>","unk_token":"<|unk|>"},"chat_template_jinja":"{% if tools is iterable and tools | length > 0 %}\n{{- '<|im_start|>system\n당신은 도구 호출 기능을 갖춘 유용한 도우미입니다. 사용자의 요청을 처리하기 위해서 필요한 도구가 주어진 목록에 있는 경우 도구 호출로 응답하세요.\n필요한 도구가 목록에 없는 경우에는 도구 호출 없이 사용자가 요구한 정보를 제공하세요.\n필요한 도구가 목록에 있지만 해당 도구를 호출하는데 필요한 argument 정보가 부족한 경우 해당 정보를 사용자에게 요청하세요.\n사용자의 요청을 처리하기 위해 여러번 도구를 호출할 수 있어야 합니다.\n도구 호출 이후 도구 실행 결과를 입력으로 받으면 해당 결과를 활용하여 답변을 생성하세요.\n다음은 접근할 수 있는 도구들의 목록입니다.<tools>\n'}}\n{% for t in tools %}\n{{- t | tojson }}\n{{- '\n'}}\n{% endfor %}\n{{- '</tools>\n\n도구를 호출하려면 아래의 JSON으로 응답하세요.\n도구 호출 형식:\n<tool_call>{{\"name\": 도구 이름, \"arguments\": dictionary 형태의 도구 인자값}}</tool_call>\n'}}\n{% endif %}\n{% macro render_content(content) -%}\n{{- '# `content` may be a plain string, none, or a list of parts as sent by\nOpenAI-compatible clients: [{\"type\": \"text\", \"text\": \"...\"}]` -#}\n{% if content is none -%}\n{% elif content is string -%}\n{{- content -}}\n{% elif content is mapping -%}\n{{- content.text if content.text is defined else content -}}\n{% elif content is sequence -%}\n{% for part in content -%}\n{% if part is mapping and part.text is defined -%}\n{{- part.text -}}\n{% elif part is string -%}\n{{- part -}}\n{% endif -%}\n{% endfor -%}\n{% else -%}\n{{- content -}}\n{% endif -%}\n{% endmacro -%}\n{% macro render_tool_calls(tool_calls) -%}\n{% for tool_call in tool_calls -%}\n{% if tool_call.function is defined -%}\n{% set tool_call = tool_call.function -%}\n{% endif -%}\n{{- '<tool_call>\n\n' -}}\n{{- '{\"name\": \"' + tool_call.name + '\"' -}}\n{% if tool_call.arguments is defined -%}\n{{- ', \"arguments\": ' -}}\n{{- '# OpenAI-compatible clients send `arguments` as a JSON string; pass it through -#}\n{% if tool_call.arguments is string -%}\n{{- tool_call.arguments -}}\n{% else -%}\n{{- tool_call.arguments | tojson -}}\n{% endif -%}\n{{- ' '\n</tool_call>' -}}\n{% endfor -%}\n{% endmacro -%}\n{% set ns = namespace(last_user_index=-1) %}\n{% for m in messages %}\n{% if m.role == 'user' %}\n{% set ns.last_user_index = loop.index0 %}\n{% endif %}\n{% endfor %}\n{% for message in messages %}\n{% if message.role == 'system' or message.role == 'user' %}\n{{- '<|im_start|>' + message.role + '\n' }}\n{% set reasoning_content = \" \" %}\n{% if message.reasoning_content is defined and message.reasoning_content is not none %}\n{% set reasoning_content = message.reasoning_content %}\n{% elif message.reasoning is defined and message.reasoning is not none %}\n{% set reasoning_content = message.reasoning %}\n{% endif %}\n{% if reasoning_content %}\n{{- '<think>' + reasoning_content.strip() + '</think>' }}\n{% endif %}\n{{- render_content(message.content) }}\n{% if message.tool_calls is defined and message.tool_calls is not none %}\n{{- render_tool_calls(message.tool_calls) }}\n{% endif %}\n{{- '<|im_end|>\n\n' }}\n{% elif message.role == 'assistant' %}\n{{- '<|im_start|>assistant\n\n' }}\n{{- '# Extract reasoning and content -#}\n{% set reasoning_content = \" \" %}\n{% set content = \" \" %}\n{% if message.content is defined and message.content is not none %}\n{% set content = render_content(message.content) %}\n{% endif %}\n{% if message.reasoning_content is defined and message.reasoning_content is not none %}\n{% set reasoning_content = message.reasoning_content %}\n{% elif message.reasoning is defined and message.reasoning is not none %}\n{% set reasoning_content = message.reasoning %}\n{% elif '</think>' in content %}\n{% set reasoning_content = content.split('</think>')[0].split('<think>')[-1] %}\n{% set content = content.split('</think>')[-1].rstrip('\n') %}\n{% endif %}\n{{- '# Determine whether to render thinking -#}\n{% set render_thinking = false %}\n{% if reasoning_content %}\n{% if show_thinking is defined and show_thinking %}\n{% set render_thinking = true %}\n{% elif not (show_thinking is defined and not show_thinking) %}\n{{- '# show_thinking undefined: show if last turn or tool-call turn after last user -#}\n{% if loop.last or (message.tool_calls is defined and loop.index0 > ns.last_user_index) %}\n{% set render_thinking = true %}\n{% endif %}\n{% endif %}\n{{- '<think>' + reasoning_content.strip() + '</think>' }}\n{% else %}\n{{- '</think>' }}\n{% endif %}\n{{- content }}\n{% if message.tool_calls is defined and message.tool_calls is not none %}\n{{- render_tool_calls(message.tool_calls) }}\n{% endif %}\n{{- '<|im_end|>' +
```


solving and a non-thinking mode for concise, low-latency responses, allowing the user to trade quality for cost on a per-request basis.

A.X K2 is developed as part of the Korean government's Sovereign AI foundation model project, aiming to build a frontier-scale model with deep understanding of the Korean language and culture.

Key Features

Large-Scale Sparse MoE (688B / 33B Active) Employs a high-capacity Mixture-of-Experts architecture (256 routed experts + 1 shared expert) that activates only 8 experts per token, enabling strong reasoning performance with practical inference efficiency. Relative to A.X K1, the number of routed experts increased from 192 to 256 while keeping the activated parameter count at 33B.

Hybrid Reasoning Control (Think / Non-Think) Supports user-controllable reasoning depth via the Think-Fusion recipe, allowing explicit multi-step reasoning or concise low-latency responses within a single unified model.

Sparse Gated Attention (SGA) for Efficient Long-Context Inference Combines a lightweight sparse-attention indexer (top- k token selection) with head-specific gated attention on top of Multi-head Latent Attention (MLA). This prunes attention computation at long context while suppressing attention sinks, delivering higher throughput and lower latency at long sequence lengths.

Gated Norm (GN) for Stable Training and Low-Precision Serving Applies an input-dependent gate immediately after RMSNorm to suppress massive activations and hidden-state outliers. This stabilizes large-scale training and produces an activation distribution with suppressed outliers, which is favorable for low-precision (FP8 / NVFP4) deployment.

Native FP8 Training and FP8 Checkpoint A.X K2 was trained from scratch natively in FP8 (MXFP8, E4M3) for both the forward and backward passes, and the released checkpoint is distributed in block-scaled FP8 (E4M3) . This halves the memory footprint relative to BF16 and enables efficient FP8 serving out of the box, without a separate post-hoc quantization step.

Tokenizer Optimized for Multilingual and Code Data Inherits the large-vocabulary BBPE-based tokenizer from A.X K1, optimized for token efficiency across five languages (English, Korean, Chinese, Japanese, and Spanish), with a strong emphasis on source code, structured text, and programming-related patterns.

Model Details

Architecture: Decoder-only Transformer with Mixture-of-Experts (MoE)

Total parameters: 688B (256 routed experts + 1 shared expert)

Active parameters: 33B per token (8 experts + 1 shared expert)

Number of layers: 61 (1 dense + 60 MoE)

Number of attention heads: 64

Hidden size: 7168

Intermediate size: 18432 (dense layer)

Expert intermediate size: 2048

Attention: Multi-head Latent Attention (MLA) with head-specific output gate and sparse attention (SGA)

Normalization: Gated Norm (gating applied after RMSNorm) with QK-normalization

Vocab size: 163,840

Context length: 262,144 tokens (256K) — 128K trained natively, extended to 256K via YaRN scaling

Training precision: Native FP8 (MXFP8, E4M3) forward and backward passes; FP32 master weights and gradients

Checkpoint precision: Block-scaled FP8 (E4M3, 128×128 weight blocks with dynamic activation scaling)

Architecture Highlights

A.X K2 is built from Gated Transformer Blocks , which integrate two gating mechanisms into the MoE transformer block: Sparse Gated Attention (SGA) in the attention sublayer and Gated Norm (GN) on the normalization path. Together they keep large-scale training stable while making long-context inference and low-precision serving efficient.

The A.X K2 transformer block. A lightweight indexer scores and ranks key–value candidates, and the selector keeps the top- k indices; MLA then performs latent attention over the selected KV cache. A head-specific output gate modulates the attention output before the output projection, while Gated Norm precedes the sublayer to smooth outliers for stable, low-precision (FP8/FP4) computation.

Sparse Gated Attention (SGA)

A.X K2 combines two mechanisms into SGA to make long-context inference efficient while keeping training stable:

Gated attention: A head-specific output gate is applied throughout pre-training on top of MLA. It introduces non-linearity into the attention output, mitigates attention sinks, and improves loss convergence.

Sparse attention: A lightweight indexer scores and ranks key–value candidates and selects a small set of top- k tokens (k = 2048) per query, substantially reducing attention compute at long sequence lengths.

The two mechanisms are mutually reinforcing: by suppressing the attention-sink mass on uninformative tokens, the output gate yields a better-calibrated signal so the indexer spends its limited budget on genuinely relevant positions. In long-context serving, as the input length grows into the tens of thousands of tokens, A.X K2 with SGA sustains substantially higher token throughput, lower time per output token (TPOT), and lower time-to-first-token (TTFT) than A.X K1.

Gated Norm (GN)

A.X K2 adopts Gated Norm , in which a learned, input-dependent gate modulates the normalized activation before it enters the residual stream. By preventing the normalized output from fitting outlier scalars, GN suppresses massive activations—hidden units orders of magnitude larger than the rest that persist across layers and are a documented source of both training instability and low-precision quantization error.

With this design, training remained sufficiently stable without the dual-normalization scheme used in A.X K1: the single post-normalization gating of Gated Norm subsumes the stabilizing effect of the earlier design while reducing architectural complexity. As an additional benefit, the resulting activation distribution—with outliers suppressed—improves block-scale utilization under narrow low-precision formats such as FP8 and NVFP4.

Think-Fusion (Hybrid Reasoning)

A.X K2 uses a single model to generate responses where reasoning before the answer can be enabled or disabled depending on usage requirements. This is achieved through the Think-Fusion recipe, which unifies thinking and non-thinking training into a single track via a paired supervised fine-tuning (SFT) dataset, followed by multi-stage on-policy reinforcement learning.

Think mode: Generates reasoning steps (<think>...</think>) before producing the answer for complex problem solving and multi-step inference.

Non-Think mode: Generates concise, direct responses optimized for low-latency usage.

Rather than a dual-track strategy, A.X K2 unifies training into a single track using a paired SFT dataset in which each prompt is associated with both a thinking and a non-thinking response, so the model learns mode control from explicit control tokens rather than from category-specific artifacts.

Training Data

A.X K2 was pre-trained on approximately 8.2 trillion tokens of web text, source code, academic papers, STEM and Q&A material, books, and synthetic data, arranged as a three-stage, difficulty-increasing curriculum (6.4T general knowledge → 1.4T high-quality reasoning → 0.36T long-context extension). The designed language mix of the first stage is English 72.7%, Korean 15.4%, code 8.3%, and Japanese, Spanish, and Chinese at roughly 1% each; later stages shift further toward English and toward reasoning, agentic, and long-context data.

Sources. English web text comes from an aggregated corpus whose largest single contributor is Nemotron-CC-v2.1 ; code comes from Nemotron-Pretraining-Code-v2 , and multilingual coverage from FineWeb2 . Korean web

text comes from an in-house crawl ($\approx 1.37\text{T}$ tokens), supplemented by an in-house PDF parsing pipeline and by synthetic data. These in-house components are proprietary and are not released. The datasets metadata above names the principal public datasets rather than an exhaustive inventory; a number of smaller additional web sources are not listed individually.

Curation. Heuristic and model-based quality filtering with deduplication, domain-aware mixture and upsampling, and difficulty scoring for curriculum learning. Documents are scored for educational value and difficulty and partitioned by a domain classifier, with quality thresholds applied per domain rather than globally and tightened as training advances; a candidate pool of roughly 16.2T tokens was reduced to the 8.2T tokens actually trained on.

Personal information. Corpora collected in-house pass an internal PII masking pipeline before training: detected identifiers — primarily email addresses and numeric identifiers such as phone numbers — are replaced with type-specific placeholder tokens (for example `<|email|>`) rather than deleted, so surrounding text is preserved, and detection rules are maintained per identifier type and reviewed for false positives. The public datasets listed above are used as released and are not re-processed, since their publishers already apply their own PII handling.

Data provenance. The source and collection method of every corpus used in training are recorded in an internal dataset inventory, and each source is reviewed for its permitted scope of use. Web data collected in-house is gathered in accordance with robots.txt directives. The public datasets listed above are used under their published licenses, and third-party corpora are acquired under agreements that license them for model training and permit release of the resulting model, while prohibiting redistribution of the underlying data itself.

Post-training. Two supervised fine-tuning stages followed by multi-task RL and safety/red-teaming RL. The SFT stages consume approximately 60.6B tokens of paired thinking / non-thinking examples across agentic tool use, mathematics, knowledge and science, code, general chat, instruction following, and safety; the subsequent RL stages run over roughly 100K prompts. Safety supervision builds on ESSA (Evolved Safety Specification Alignment), which conditions data synthesis on evolved domain–task safety specifications and rewards safe completion over blanket refusal.

See the technical report for the full data pipeline, mixture tables, and post-training recipe.

Evaluation Results

Thinking Mode

All models are evaluated in thinking mode . Benchmarks marked with an asterisk (*) are evaluated by Artificial Analysis . GDPval is reported as an Elo rating; all other benchmarks are reported as percentages. The best score in each row is shown in bold . See the technical report for evaluation details.

Domain Benchmark A.X K2 (688B-A33B) A.X K1 (519B-A33B) Qwen3.5 (397B-A17B) Nemotron 3 Ultra
DeepSeek-V4 Flash GLM-5.1 Kimi-K2.6 MiniMax M2.7

Math AIME26 97.1 91.3 92.5 85.4 96.7 95.4 95.4 88.8

Apex 45.8 1.0 13.5 3.1 28.1 10.4 20.8 1.0

Apex-shortlist 88.6 50.8 61.2 45.5 88.0 72.1 72.6 30.3

KMO26 (1st round) 92.5 85.0 78.8 85.0 94.4 78.1 88.1 80.0

Korean KMMLU-Pro 80.5 68.9 78.4 69.2 78.8 76.5 73.4 66.6

KoBALT 73.0 51.0 69.9 59.1 75.3 72.0 66.0 51.4

CLiCk 91.6 85.3 88.4 84.1 89.6 88.8 80.9 76.1

Code LiveCodeBench v6 (Feb–May) 84.0 74.9 82.6 76.4 89.4 86.4 86.5 69.8

SciCode 41.0 24.1 42.0 39.9 44.9 43.8 53.5 47.0

Terminal Bench v2.1* 36.0 – 51.3 53.9 61.8 61.8 65.9 55.4

Science & Knowledge Humanity’s Last Exam 27.8 8.3 27.3 26.6 32.1 28.0 35.9 28.1

GPQA Diamond 85.6 76.4 89.3 86.7 89.4 86.8 91.1 87.4

AA-Omniscience* 39.6 15.5 24.3 38.3 26.0 39.7 42.3 39.3

General IFBench 75.9 63.2 78.8 81.4 79.2 76.3 76.0 75.7

AA-LCR 66.0 26.0 65.7 67.0 63.0 62.3 69.7 68.7

Agentic GDPval* (Elo) 1031 500 962 1164 1189 1257 1190 1158

τ^2 -Bench* (Telecom) 98.0 86.0 95.6 83.3 95.0 97.7 95.9 84.8

τ^3 -Bench* (Banking) 13.0 – 13.4 13.8 22.9 11.5 20.6 8.9

BrowseComp (≤ 10 searches) 9.3 – 26.9 13.4 16.8 29.1 21.5 14.2

A.X K2 is strongest in the Math and Korean domains, achieving the best scores among all compared models on AIME26, Apex, Apex-shortlist, KMMLU-Pro, and CLiCK. Beyond final-answer benchmarks, A.X K2 scored 35/42 on IMO 2025 —reaching the gold-medal threshold with a perfect 7/7 on each of the first five problems—and produced correct proofs for all 8 problems of the KMO26 2nd round , using an iterative proof refinement method. It also achieves the best score on τ^2 -Bench Telecom among all compared models.

Long Context

A.X K2 is trained natively to a 128K context via Adjusted Base Frequency (ABF) and extends to longer sequences through zero-shot YaRN scaling at inference. We assess long-context capability along two complementary axes—comprehensive understanding on RULER and long-range fact retrieval via needle-in-a-haystack (NIAH) . See the technical report for details.

RULER

RULER measures long-context understanding across retrieval, multi-hop tracing, aggregation, and question-answering tasks at increasing sequence lengths. A.X K2 sustains strong performance out to 256K, achieving an overall average of 94.6.

Model 4K 8K 16K 32K 64K 128K 256K Overall

A.X K2 97.5 97.2 97.5 96.5 94.3 92.7 86.6 94.6

Needle-in-a-Haystack (NIAH)

NIAH probes exact fact retrieval by inserting a target fact (the "needle") at varying depths within a long context and asking the model to recover it. Under zero-shot YaRN scaling (scaling factors of 1, 2, and 4 for 128K, 256K, and 512K, respectively), A.X K2 attains a perfect retrieval score at every context length and needle depth—and does so even after NVFP4 (experts-only W4A4) quantization. The heatmaps below show the 256K and 512K results.

NIAH fact retrieval across context length (x-axis) and needle depth (y-axis): 256K (YaRN factor 2, top) and 512K (YaRN factor 4, bottom), under NVFP4 (experts-only W4A4) quantization. A.X K2 scores a perfect 100 at every position.

Usage

Transformers

A.X K2 is natively supported in Hugging Face Transformers . The axk2 architecture ships with the library itself, so neither `trust_remote_code=True` nor any custom modeling file is needed — see the axk2 model documentation .

The integration was merged after the v5.14.1 release, so until the next release is published, install Transformers from source. FP8 execution additionally needs the kernels package, which supplies the block-scaled FP8 matmul the checkpoint runs on:

```
# install torch first, built against your CUDA version
pip install git+https://github.com/huggingface/transformers.git
accelerate "kernels>=0.15.2,<0.16.0"
```

```
from transformers import AutoModelForCausalLM, AutoTokenizer
```

```
model_name = "skt/A.X-K2"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name, device_map="auto")
```

```

messages = [{"role": "user", "content": "A.X K2에 대해 간단히 소개해줘."}] inputs =
tokenizer.apply_chat_template( messages, add_generation_prompt= True , enable_thinking= True , # omit or set
False for non-thinking mode return_tensors= "pt" , return_dict= True , ).to(model.device)

outputs = model.generate(**inputs, max_new_tokens= 1024 ) print (tokenizer.decode(outputs[ 0 ][inputs[


```

In thinking mode the output begins with the reasoning trace and closes it with `</think>` . These two markers are ordinary tokens rather than special tokens, so they survive decoding and can be used to split the reasoning from the final answer.

FP8 execution is configured by the checkpoint itself (`quantization_config` in `config.json`), so no separate quantization step is needed. It requires `accelerate` , the `kernels` package, and a GPU of compute capability 8.9 or higher (for example RTX 4090, H100, B200); on SM90 and newer, DeepGEMM kernels are used when available and the `kernels Triton` path serves as the fallback. On older GPUs — or when no GPU is visible — Transformers dequantizes the weights to `bfloat16` automatically, which needs roughly 1.4 TB of memory for the full model.

The FP8 weights are about 647 GiB on disk and take a little more than that once loaded, so plan on at least 800 GiB of aggregate GPU memory with the KV cache and activations included — four B300 cards (275 GiB each) were used for validation, and twelve or more 80 GB cards would be needed instead.

Under Transformers, A.X K2 runs with the `sdpa` (default and recommended) and `eager` attention implementations; `flash_attention_2` and `flex_attention` are not supported. This path applies the sparse-attention selection as a dense additive mask, so use the vLLM path below for long context and for serving, where the indexer's top- k indices reach the sparse MLA kernels directly.

For runnable scripts and the operational detail this section leaves out — hardware sizing, the FP8 kernel pre-check, air-gapped kernel setup, and a chat-template test suite — see <https://github.com/SKT-AI/A.X-K2> .

vLLM

A.X K2 runs on vLLM through our fork at `SKT-AI/vllm` , which is upstream vLLM v0.23.0 plus A.X K2 support and nothing else. Upstreaming is in preparation, and this section will be updated with the pull request link once it is submitted.

```

git clone -b axk2-v0.23.0 https://github.com/SKT-AI/vllm.git cd vllm pip install -e . # see the vLLM docs for build
prerequisites

```

The `axk2` config and model are registered in-tree on that branch, so `--trust-remote-code` is not needed:

```

vllm serve skt/A.X-K2 --tensor-parallel-size <N> \ --tool-call-parser hermes \ --reasoning-parser deepseek_v3

```

The two parser flags are what turn tool calls and the thinking trace into structured fields on the OpenAI-compatible endpoint. `deepseek_v3` reads `enable_thinking` out of each request's `chat_template_kwargs` , so the mode is chosen per call; use it rather than `deepseek_r1` , which would classify a whole non-thinking answer as reasoning.

Loaded, the FP8 weights occupy roughly 656 GiB of aggregate GPU memory spread evenly across the tensor-parallel ranks, so choose `<N>` such that the node holds that much with room left for the KV cache; the reference run used four 275 GB cards.

The full 256K context needs no extra flags — the YaRN configuration ships in `config.json` — and needle-in-a-haystack retrieval was verified at 32K, 128K, and 256K. This path feeds the indexer's top- k indices straight to the sparse MLA kernels (FlashMLA / FlashInfer) instead of building a dense attention mask, which is what makes the sparse attention pay off at length.

The first launch is dominated by one-time kernel work — DeepGEMM warmup and FlashInfer JIT compilation — rather than by reading weights, so budget for it and keep `VLLM_CACHE_ROOT` on persistent storage; see the example repository for the startup timings and prerequisites.

The branch is pinned to a released vLLM tag rather than tracking main ; newer bases will be published as additional branches.

Intended Use

A.X K2 is a general-purpose, text-only foundation model released as open weights under Apache-2.0 for both research and commercial use. Built as an agentic foundation model, it is intended for multi-step reasoning and tool use, Korean- and English-language assistant applications, software engineering, mathematics and knowledge-intensive question answering, long-context workloads up to 256K tokens, and further adaptation as a base model. Use thinking mode where accuracy matters more than latency, and non-thinking mode for latency- or cost-sensitive use.

As part of the Korean government's Sovereign AI foundation model project, A.X K2 is intended in particular to enable controllable, self-hosted AI adoption across Korean industry, government, and academia.

A.X K2 is a foundation model, not a finished application. Deployments should keep a human in the loop for consequential decisions and add application-level safeguards appropriate to their domain and jurisdiction.

Out-of-Scope Use

A.X K2 has not been evaluated or certified for autonomous decision-making in high-stakes domains. It should not be the sole or primary basis for medical, financial-eligibility, employment, legal, law-enforcement, or safety-critical decisions.

We also ask users to refrain from using A.X K2 to generate unlawful content or material that infringes third-party rights, to produce targeted disinformation or impersonate real people or organizations, to circumvent its safety behavior for operational harm, or to process personal data without a lawful basis.

The model is text-only: image, audio, and video inputs are not supported. Its tokenizer and training data target five languages (English, Korean, Chinese, Japanese, and Spanish); use in other languages is untested.

Limitations and Biases

Factual reliability. A.X K2 may generate incorrect or misleading information due to its stochastic nature; verify outputs before relying on them.

Modality. Text-only; no native multimodal understanding.

Language coverage. Pre-training data is predominantly English (roughly 73–90% by stage) and Korean (roughly 6–15%), with Chinese, Japanese, and Spanish at about 1% each, so quality in the latter three is materially less validated. Being largely web text, the corpus also leaves knowledge of long-tail regional and specialist topics uneven.

Bias. The model can reproduce social biases present in web-scale data, including stereotypes relating to gender, age, region, occupation, and nationality. Bias is one of nine dimensions judged during safety RL, but we did not run a dedicated quantitative bias or fairness evaluation — evaluate on your own population and task before deploying in bias-sensitive contexts.

Safety. Safety alignment reduces but does not eliminate unsafe outputs; adversarial prompting may still elicit harmful content.

Agentic capability. Agentic performance is moderate — A.X K2 trails the strongest compared models on Terminal Bench and BrowseComp — reflecting limited agentic RL during post-training. Evaluate long-horizon tool-use workloads before relying on them.

Evaluation coverage. Reported benchmarks are predominantly English and Korean, and largely single-turn.

Lightweight Models

If you are interested in using lightweight versions of A.X K2 for business applications or scalable deployment, please contact us at a.x@sk.com.

Contact

For questions about A.X K2 — including model behavior, deployment, and licensing — contact the A.X team at a.x@sk.com. Please send reports of vulnerabilities, harmful outputs, suspected misuse, or copyright infringement claims to the same address.

Citation

If you use A.X K2 in your research, please cite the technical report:

@techreport{baek2026axk2technicalreport, title={A.X K2 Technical Report}, author={Cheolseung Baek and Dhammiko Arya and Eunki Kim and Gun Song and Gyoungyeon Han and Hyunho Yang and Hyunjun Eun and Jin Kim and Junyoung Park and Juyun Wee and Minki Hong and Minkyung Park and Minsang Kim and Minsoo Kang and SaeRom Kim and Sangjin Kim and Sangyeol Lee and Seojin Lee and Seokhwan Jo and Seokyoung Hong and Seongho Choi and Seonghye Cho and Seongmin Ok and Sereimony Sek and Seungmo Cho and Seungsik Kim and Singon Kim and Sohee Park and Sooyeon Park and Subin Yi and Sungbin Yoon and Sungeun Lee and Sung Jun Cheon and Sungwan Kim and Sunwoo Lee and Tae Yoon Kim and Wonbeom Jang and Yohan Ra and Yong-jin Han and Youngjin Kim and Youngrang Kim and Yujin Kang and Yujin Lee}, year={2026}, institution={SK Telecom}, eprint={2608.30181}, archivePrefix={arXiv}, primaryClass={cs.AI}, url={https://arxiv.org/abs/2608.30181}, }

Downloads last month 17,096

```
<|cls>">","eos_token":"</im_end">,"mask_token":"","pad_token":"","sep_token":"<|sep>","unk_token":"<|unk>"},"chat_template_jinja":"%- if tools is iterable and tools | length > 0 %}\n{- '<|im_start>'>\nsystem\n당신은 도구 호출 기능을 갖춘 유용한 도우미입니다. 사용자의 요청을 처리하기 위해서 필요한  
도구가 주어진 목록에 있는 경우 도구 호출로 응답하세요.\n필요한 도구가 목록에 없는 경우에는 도구 호출 없이 사  
용자가 요구한 정보를 제공하세요.\n필요한 도구가 목록에 있지만 해당 도구를 호출하는데 필요한 argument 정보  
가 부족한 경우 해당 정보를 사용자에게 요청하세요.\n사용자의 요청을 처리하기 위해 여러번 도구를 호출할 수 있  
어야 합니다.\ndo구 호출 이후 도구 실행 결과를 입력으로 받으면 해당 결과를 활용하여 답변을 생성하세요.\nn다  
음은 접근할 수 있는 도구들의 목록 입니다:\n<tools>\n' }\n %- for t in tools %}\n {- {t | toJSON} }\n {\n %-  
endfor %}\n{\n'- />'\ndo구를 호출하려면 아래의 JSON으로 응답하세요.\ndo구 호출 형식:  
  
<tool_call>{'name': 도구 이름, 'arguments': dictionary 형태의 도구 인자값}</tool_call><|im_end>\n'} }\n %-  
endif %}\n\ndef macro render_content(content) -%}\n #- `content` may be a plain string, none, or a list of parts as  
sent by\nOpenAI-compatible clients: [{"type": "text", "text": "..."}] -#}\n %- if content is none -%}\n {\n %- elif  
content is string -%}\n {{- content -}}}\n %- elif content is mapping -%}\n {{- content.text if content.text is defined  
else content -}}}\n %- elif content is sequence -%}\n %- for part in content -%}\n %- if part is mapping and  
part.text is defined -%}\n {{- part.text -}}}\n %- elif part is string -%}\n {{- part -}}}\n %- endif -%}\n %- endfor -%}\n \n %- else -%}\n {{- content -}}}\n %- endif -%}\n{%- endmacro -%}\n\ndeclare ns namespace(last_user_index=-1)\n %- for m in messages %}\n %- if m.role ==  
'user' %}\n %- set ns.last_user_index = loop.index0 %}\n %- endif %}\n %- endfor %}\n\ncurrent_message message in  
messages %}\n %- if message.role == 'system' or message.role == 'user' %}\n {{- '<|im_start>' + message.role +  
'\n' }}\n %- set reasoning_content = "" %}\n %- if message.reasoning_content is defined and  
message.reasoning_content is not none %}\n {%- set reasoning_content = message.reasoning_content %}\n %-  
elif message.reasoning is defined and message.reasoning is not none %}\n {%- set reasoning_content =  
message.reasoning %}\n %- endif %}\n %- if reasoning_content %}\n {{- '<think>' + reasoning_content.strip() +  
'</think>' }}\n %- endif %}\n {{- render_content(message.content) }}\ntool_calls if message.tool_calls is defined and  
message.tool_calls is not none %}\n {{- render_tool_calls(message.tool_calls) }}\n %- endif %}\n {{- '<|im_end>'  
\n'}}\n %- elif message.role == 'assistant' %}\n {{- '<|im_start>assistant\n' }}\n #- Extract reasoning and  
content -#\n %- set reasoning_content = "" %}\n %- set content = "" %}\n %- if message.content is defined and  
message.content is not none %}\n %- set content = render_content(message.content) %}\n %- endif %}\n %- if  
message.reasoning_content is defined and message.reasoning_content is not none %}\n %- set  
reasoning_content = message.reasoning_content %}\n %- elif message.reasoning is defined and  
message.reasoning is not none %}\n {%- set reasoning_content = message.reasoning %}\n %- elif '</think>' in  
content %}\n %- set reasoning_content = content.split('</think>')[0].split('<think>')[-1] %}\n %- set content =  
content.split('</think>')[-1].rstrip("\n") %}\n %- endif %}\n #- Determine whether to render thinking -#\n %- set  
render_thinking = false %}\n %- if reasoning_content %}\n %- if show_thinking is defined and show_thinking  
%}\n %- set render_thinking = true %}\n %- elif not (show_thinking is defined and not show_thinking) %}\n #-  
show_thinking undefined: show if last turn or tool-call turn after last user -#\n %- if loop.last or  
(message.tool_calls is defined and loop.index0 > ns.last_user_index) %}\n %- set render_thinking = true %}\n %-  
endif %}\n %- endif %}\n %- endif %}\n %- endif %}\n %- if render_thinking %}\n {{- '<think>' + reasoning_content.strip() + '  

```

```

think>' }}\n {%-% else %}\n {{- '</think>' }}\n {%-% endif %}\n\n {{- content }}\n\n{{%- if message.tool_calls is defined and
message.tool_calls is not none %}\n {{- render_tool_calls(message.tool_calls) }}\n {%-% endif %}\n {{- '<|im_end|>' +
('\n' if (not loop.last or add_generation_prompt) else '') }}\n {%-% elif message.role == 'tool' %}\n {{- '<|im_start|>tool\
\n' }}\n {{- '<tool_response>\n' }}\n {{- render_content(message.content) }}\n {{- '\n' }}\n {{- '</tool_response><|
im_end|>\n' }}\n {%-% endif %}\n{%-% endfor %}\n\n{{%- if add_generation_prompt %}\n {{- '<|im_start|>assistant\n'
}}\n {%-% if enable_thinking is defined and enable_thinking is true %}\n {{- '<think>' }}\n {%-% else %}\n {{- '</think>'
}}\n {%-% endif %}\n{%-% endif
%}\n"},"createdAt":"2026-07-28T10:33:49.000Z","discussionsDisabled":false,"discussionsSorting":"recently-
created","downloads":17096,"downloadsAllTime":156524,"id":"skt/A.X-
K2","isLikedByUser":false,"availableInferenceProviders":
[],"showHuggingChatEntry":false,"inference":"","lastModified":"2026-09-01T06:57:43.000Z","likes":107,"pipeline_tag":
"generation","library_name":"transformers","librariesOther":[],"trackDownloads":true,"mask_token":"<|mask|
>","model-index":null,"evalResults":{"dataset":{"id":"Idavidrein/
gpqa","isBenchmark":true,"task_id":"diamond"},"value":85.6,"source":{"url":"https://huggingface.co/skt/A.X-K2/
blob/fb412dcd22e9c5ac43b2967d6c5e8a7d0e667b0f/README.md#L134-L152","name":"Model
Card","isExternal":false,"filename":"eval_results/gpqa.yaml","verified":false,"rank":44,"label":"Diamond"},
{"dataset":{"id":"MathArena/aime_2026","isBenchmark":true,"task_id":"MathArena/
aime_2026"},"value":97.1,"source":{"url":"https://huggingface.co/skt/A.X-K2/blob/
fb412dcd22e9c5ac43b2967d6c5e8a7d0e667b0f/README.md#L134-L152","name":"Model
Card","isExternal":false,"filename":"eval_results/aime_2026.yaml","verified":false,"rank":4,"label":"MathArena
Aime 2026"},"dataset":{"id":"cais/hle","isBenchmark":false,"task_id":"hle"},"value":27.8,"source":{"url":"https://
huggingface.co/skt/A.X-K2/blob/fb412dcd22e9c5ac43b2967d6c5e8a7d0e667b0f/README.md#L134-
L152","name":"Model Card","isExternal":false,"filename":"eval_results/
hle.yaml","verified":false,"rank":41,"label":"Hle"},"private":false,"repoType":"model","gated":false,"tags":
["transformers","safetensors","axk2","text-generation","conversational","en","ko","zh","ja","es","dataset:nvidia/
Nemotron-CC-v2.1","dataset:nvidia/Nemotron-Pretraining-Code-v2","dataset:HuggingFaceFW/
fineweb-2","arxiv:2608.30181","license:apache-2.0","eval-
results","endpoints_compatible","fp8","region:us"],"tag_objs":{"id":"text-generation","label":"Text
Generation","type":"pipeline_tag","subType":"nlp"},"id":"transformers","label":"Transformers","type":"library"},
{"id":"safetensors","label":"Safetensors","type":"library"},"id":"dataset:nvidia/Nemotron-CC-v2.1","label":"nvidia/
Nemotron-CC-v2.1","type":"dataset","extra":{"disabled":false},"id":"dataset:nvidia/Nemotron-Pretraining-Code-
v2","label":"nvidia/Nemotron-Pretraining-Code-v2","type":"dataset","extra":{"disabled":false}},
{"id":"dataset:HuggingFaceFW/fineweb-2","label":"HuggingFaceFW/fineweb-2","type":"dataset","extra":
{"disabled":false},"id":"en","label":"English","type":"language"},"id":"ko","label":"Korean","type":"language"},
{"id":"zh","label":"Chinese","type":"language"},"id":"ja","label":"Japanese","type":"language"},
{"id":"es","label":"Spanish","type":"language"},"id":"axk2","label":"axk2","type":"other","clickable":true},
{"id":"conversational","label":"conversational","type":"other","clickable":true},"id":"eval-results","label":"Eval
Results","type":"other","clickable":true},"id":"endpoints_compatible","label":"Inference
Endpoints","type":"other","clickable":true},"id":"fp8","label":"fp8","type":"other","clickable":true},
{"id":"arxiv:2608.30181","label":"arxiv:2608.30181","type":"arxiv","extra":{"paperTitle":"A.X K2 Technical Report"}},
{"id":"license:apache-2.0","label":"apache-2.0","type":"license"},"type":"region","label":"🌐 Region:
US","id":"region:us"},"transformersInfo":{"auto_model":"AutoModelForCausalLM","pipeline_tag":"text-
generation","processor":"AutoTokenizer"},"widgetData":{"text":"Hi, what can you help me with?","text":"What is
84 * 3 / 2?","text":"Tell me an interesting fact about the universe!","text":"Explain quantum computing in simple
terms."},"safetensors":{"parameters":
{"F32":15360,"BF16":2515671296,"F8_E4M3":689174544384,"total":691690231040,"sharded":true,"totalFileSize":61
{"billableEntities":[],"entityName2Providers":{}},"canWrite":false}> Safetensors Model size 692B params

```

Tensor type F32 · BF16 · F8_E4M3 ·

Chat template Files info

```

", "cls_token": "<|cls|>", "eos_token": "<|im_end|>", "mask_token": "<|mask|>", "pad_token": "<|pad|>", "sep_token": "<|
sep|>", "unk_token": "<|unk|>"},"chat_template_jinja":"{{%- if tools is iterable and tools | length > 0 %}}\n{{- '<|
im_start|>system\n당신은 도구 호출 기능을 갖춘 유용한 도우미입니다. 사용자의 요청을 처리하기 위해서 필요한
도구가 주어진 목록에 있는 경우 도구 호출로 응답하세요.\n필요한 도구가 목록에 없는 경우에는 도구 호출 없이 사
용자가 요구한 정보를 제공하세요.\n필요한 도구가 목록에 있지만 해당 도구를 호출하는데 필요한 argument 정보
가 부족한 경우 해당 정보를 사용자에게 요청하세요.\n사용자의 요청을 처리하기 위해 여러번 도구를 호출할 수 있

```

어야 합니다.\n도구 호출 이후 도구 실행 결과를 입력으로 받으면 해당 결과를 활용하여 답변을 생성하세요.\n\n다음은 접근할 수 있는 도구들의 목록입니다:\n<tools>\n' }\n {% for t in tools %}\n {{ t | tojson }}\n {{ '\n' }}\n {% endfor %}\n {{ '</tools>\n\n도구를 호출하려면 아래의 JSON으로 응답하세요.\n도구 호출 형식:

```

<tool_call>{"name": 도구 이름, "arguments": dictionary 형태의 도구 인자값}</tool_call><|im_end|>\n' }\n {% endfor %}\n\n{% macro render_content(content) -%}\n {%# `content` may be a plain string, none, or a list of parts as sent by\n OpenAI-compatible clients: [{\"type\": \"text\", \"text\": \"...\"}] -#}\n {% if content is none -%}\n {% elif content is string -%}\n {{ content }}\n {% elif content is mapping -%}\n {{ content.text if content.text is defined else content }}\n {% elif content is sequence -%}\n {% for part in content -%}\n {% if part is mapping and part.text is defined -%}\n {{ part.text }}\n {% elif part is string -%}\n {{ part }}\n {% endif -%}\n {% endfor -%}\n {% else -%}\n {{ content }}\n {% endif -%}\n {% endmacro -%}\n\n{% macro render_tool_calls(tool_calls) -%}\n {% for tool_call in tool_calls -%}\n {% if tool_call.function is defined -%}\n {% set tool_call = tool_call.function -%}\n {% endif -%}\n {{ '<tool_call>\n' }}\n {{ '{\"name\": \"' + tool_call.name + '\"' }}\n {% if tool_call.arguments is defined -%}\n {{ ', \"arguments\": ' }}\n {%# OpenAI-compatible clients send `arguments` as a JSON string; pass it through -#}\n {% if tool_call.arguments is string -%}\n {{ tool_call.arguments }}\n {% else -%}\n {{ tool_call.arguments | tojson }}\n {% endif -%}\n {{ '\n</tool_call>' }}\n {% endfor -%}\n\n{% endmacro -%}\n\n{% set ns = namespace(last_user_index=-1) %}\n {% for m in messages %}\n {% if m.role == 'user' %}\n {% set ns.last_user_index = loop.index0 %}\n {% endif %}\n {% endfor %}\n\n{% for message in messages %}\n {% if message.role == 'system' or message.role == 'user' %}\n {{ '<|im_start|>' + message.role + '\n' }}\n {% set reasoning_content = \" \" %}\n {% if message.reasoning_content is defined and message.reasoning_content is not none %}\n {% set reasoning_content = message.reasoning_content %}\n {% elif message.reasoning is defined and message.reasoning is not none %}\n {% set reasoning_content = message.reasoning %}\n {% endif %}\n {% if reasoning_content %}\n {{ '<think>' + reasoning_content.strip() + '</think>' }}\n {% endif %}\n {{ render_content(message.content) }}\n\n{% if message.tool_calls is defined and message.tool_calls is not none %}\n {{ render_tool_calls(message.tool_calls) }}\n {% endif %}\n {{ '<|im_end|>\n' }}\n\n{% elif message.role == 'assistant' %}\n {{ '<|im_start|>assistant\n' }}\n\n{%# Extract reasoning and content -#\n {% set reasoning_content = \" \" %}\n {% set content = \" \" %}\n {% if message.content is defined and message.content is not none %}\n {% set content = render_content(message.content) %}\n {% endif %}\n {% if message.reasoning_content is defined and message.reasoning_content is not none %}\n {% set reasoning_content = message.reasoning_content %}\n {% elif message.reasoning is defined and message.reasoning is not none %}\n {% set reasoning_content = message.reasoning %}\n {% endif %}\n {% if '<think>' in content %}\n {% set reasoning_content = content.split('</think>')[0].split('<think>')[1] %}\n {% set content = content.split('</think>')[1].lstrip('\n') %}\n {% endif %}\n\n{%# Determine whether to render thinking -#\n {% set render_thinking = false %}\n {% if reasoning_content %}\n {% if show_thinking is defined and show_thinking %}\n {% set render_thinking = true %}\n {% elif not (show_thinking is defined and not show_thinking) %}\n {%# show_thinking undefined: show if last turn or tool-call turn after last user -#\n {% if loop.last or (message.tool_calls is defined and loop.index0 > ns.last_user_index) %}\n {% set render_thinking = true %}\n {% endif %}\n {% endif %}\n {{ '<think>' + reasoning_content.strip() + '</think>' }}\n {% else %}\n {{ '</think>' }}\n {% endif %}\n\n{{ content }}\n\n{% if message.tool_calls is defined and message.tool_calls is not none %}\n {{ render_tool_calls(message.tool_calls) }}\n {% endif %}\n {{ '<|im_end|>' + ('\n' if (not loop.last or add_generation_prompt) else \" \" ) }}\n\n{% elif message.role == 'tool' %}\n {{ '<|im_start|>tool\n' }}\n {{ '<tool_response>\n' }}\n {{ render_content(message.content) }}\n {{ '\n' }}\n {{ '</tool_response><|im_end|>\n' }}\n\n{% endif %}\n {% endfor %}\n\n{% if add_generation_prompt %}\n {{ '<|im_start|>assistant\n' }}\n\n{% if enable_thinking is defined and enable_thinking is true %}\n {{ '<think>' }}\n {% else %}\n {{ '</think>' }}\n {% endif %}\n\n{% endfor %}\n\n{% endmacro -%}\n\n{%}\n\n\", \"createdAt\": \"2026-07-28T10:33:49.000Z\", \"discussionsDisabled\": false, \"discussionsSorting\": \"recently-created\", \"downloads\": 17096, \"downloadsAllTime\": 156524, \"id\": \"skt/A.X-K2\", \"isLikedByUser\": false, \"availableInferenceProviders\":\n\n[], \"showHuggingChatEntry\": false, \"inference\": \"\", \"lastModified\": \"2026-09-01T06:57:43.000Z\", \"likes\": 107, \"pipeline_tag\": \"generation\", \"library_name\": \"transformers\", \"librariesOther\": [], \"trackDownloads\": true, \"mask_token\": \"<|mask|>\", \"model-index\": null, \"evalResults\": [{\"dataset\": {\"id\": \"ldavidrein/gpqa\", \"isBenchmark\": true, \"task_id\": \"diamond\", \"value\": 85.6, \"source\": {\"url\": \"https://huggingface.co/skt/A.X-K2/blob/fb412dcd22e9c5ac43b2967d6c5e8a7d0e667b0f/README.md#L134-L152\", \"name\": \"Model Card\", \"isExternal\": false, \"filename\": \"eval_results/gpqa.yaml\", \"verified\": false, \"rank\": 44, \"label\": \"Diamond\"}, {\"dataset\": {\"id\": \"MathArena/aime_2026\", \"isBenchmark\": true, \"task_id\": \"MathArena/aime_2026\", \"value\": 97.1, \"source\": {\"url\": \"https://huggingface.co/skt/A.X-K2/blob/fb412dcd22e9c5ac43b2967d6c5e8a7d0e667b0f/README.md#L134-L152\", \"name\": \"Model

```

```
Card", "isExternal": false, "filename": ".eval_results/aime_2026.yaml", "verified": false, "rank": 4, "label": "MathArena  
Aime 2026"}, {"dataset": {"id": "cais/hle", "isBenchmark": false, "task_id": "hle", "value": 27.8, "source": {"url": "https://  
huggingface.co/skt/A.X-K2/blob/fb412dcd22e9c5ac43b2967d6c5e8a7d0e667b0f/README.md#L134-  
L152", "name": "Model Card", "isExternal": false}, "filename": ".eval_results/  
hle.yaml", "verified": false, "rank": 41, "label": "Hle"}}, {"private": false, "repoType": "model", "gated": false, "tags":  
["transformers", "safetensors", "axk2", "text-generation", "conversational", "en", "ko", "zh", "ja", "es", "dataset:nvidia/  
Nemotron-CC-v2.1", "dataset:nvidia/Nemotron-Pretraining-Code-v2", "dataset:HuggingFaceFW/  
fineweb-2", "arxiv:2608.30181", "license:apache-2.0", "eval-  
results", "endpoints_compatible", "fp8", "region:us"], "tag_objs": [{"id": "text-generation", "label": "Text  
Generation", "type": "pipeline_tag", "subType": "nlp"}, {"id": "transformers", "label": "Transformers", "type": "library"},  
{ "id": "safetensors", "label": "Safetensors", "type": "library"}, {"id": "dataset:nvidia/Nemotron-CC-v2.1", "label": "nvidia/  
Nemotron-CC-v2.1", "type": "dataset", "extra": {"disabled": false}}, {"id": "dataset:nvidia/Nemotron-Pretraining-Code-  
v2", "label": "nvidia/Nemotron-Pretraining-Code-v2", "type": "dataset", "extra": {"disabled": false}},  
{ "id": "dataset:HuggingFaceFW/fineweb-2", "label": "HuggingFaceFW/fineweb-2", "type": "dataset", "extra":  
{ "disabled": false}}, {"id": "en", "label": "English", "type": "language"}, {"id": "ko", "label": "Korean", "type": "language"},  
{ "id": "zh", "label": "Chinese", "type": "language"}, {"id": "ja", "label": "Japanese", "type": "language"},  
{ "id": "es", "label": "Spanish", "type": "language"}, {"id": "axk2", "label": "axk2", "type": "other", "clickable": true},  
{ "id": "conversational", "label": "conversational", "type": "other", "clickable": true}, {"id": "eval-results", "label": "Eval  
Results", "type": "other", "clickable": true}, {"id": "endpoints_compatible", "label": "Inference  
Endpoints", "type": "other", "clickable": true}, {"id": "fp8", "label": "fp8", "type": "other", "clickable": true},  
{ "id": "arxiv:2608.30181", "label": "arxiv:2608.30181", "type": "arxiv", "extra": {"paperTitle": "A.X K2 Technical Report"}},  
{ "id": "license:apache-2.0", "label": "apache-2.0", "type": "license"}, {"type": "region", "label": "🌐 Region:  
US", "id": "region:us"}}, {"transformersInfo": {"auto_model": "AutoModelForCausalLM", "pipeline_tag": "text-  
generation", "processor": "AutoTokenizer"}, "widgetData": [{"text": "Hi, what can you help me with?"}, {"text": "What is  
84 * 3 / 2?"}, {"text": "Tell me an interesting fact about the universe!"}, {"text": "Explain quantum computing in simple  
terms."}], "safetensors": {"parameters":  
{ "F32": 15360, "BF16": 2515671296, "F8_E4M3": 689174544384, "total": 691690231040, "sharded": true, "totalFileSize": 691690231040},  
{"inferenceContextData": {"billableEntities": [], "entityName2Providers": {}}, "linkToInferenceDiscussion": {"url": "/  
spaces/huggingface/InferenceSupport/  
discussions/11355", "numReactions": 7}, "isLoggedIn": false, "canWrite": false, "examples":  
[], "widgetType": "conversational", "isMaximized": false, "isMac": true}}> Inference Providers NEW
```

Text Generation

This model isn't deployed by any Inference Provider. 🤖 7 Ask for provider support

Model tree for skt/A.X-K2

Finetunes

2 models

Quantizations

2 models

Datasets used to train skt/A.X-K2

Space using skt/A.X-K2 1

Collections including skt/A.X-K2 4 items • Updated Aug 9 • 15

8 items • Updated Aug 22 • 7

Paper for skt/A.X-K2 Paper • 2608.30181 • Published Aug 31

```
"cls_token": "<|cls|>", "eos_token": "<|im_end|>", "mask_token": "<|mask|>", "pad_token": "<|pad|>", "sep_token": "<|  
sep|>", "unk_token": "<|unk|>", "chat_template_jinja": "{%- if tools is iterable and tools | length > 0 %}\n{{- '<|  
im_start|>system\n당신은 도구 호출 기능을 갖춘 유용한 도우미입니다. 사용자의 요청을 처리하기 위해서 필요한  
도구가 주어진 목록에 있는 경우 도구 호출로 응답하세요.\n필요한 도구가 목록에 없는 경우에는 도구 호출 없이 사  
용자가 요구한 정보를 제공하세요.\n필요한 도구가 목록에 있지만 해당 도구를 호출하는데 필요한 argument 정보  
가 부족한 경우 해당 정보를 사용자에게 요청하세요.\n사용자의 요청을 처리하기 위해 여러번 도구를 호출할 수 있
```

어야 합니다.\n도구 호출 이후 도구 실행 결과를 입력으로 받으면 해당 결과를 활용하여 답변을 생성하세요.\n\n다음은 접근할 수 있는 도구들의 목록입니다:\n<tools>\n' }\n {% for t in tools %}\n {{ t | tojson }}\n {{ '\n' }}\n {% endfor %}\n {{ '</tools>\n\n도구를 호출하려면 아래의 JSON으로 응답하세요.\n도구 호출 형식:

```

<tool_call>{"name": "도구 이름", "arguments": dictionary 형태의 도구 인자값}</tool_call><|im_end|>\n' }\n {% endfor %}\n\n{% macro render_content(content) -%}\n {%# `content` may be a plain string, none, or a list of parts as sent by\n OpenAI-compatible clients: [{\"type\": \"text\", \"text\": \"...\"}] -#}\n {% if content is none -%}\n {% elif content is string -%}\n {{ content }}\n {% elif content is mapping -%}\n {{ content.text if content.text is defined else content }}\n {% elif content is sequence -%}\n {% for part in content -%}\n {% if part is mapping and part.text is defined -%}\n {{ part.text }}\n {% elif part is string -%}\n {{ part }}\n {% endif -%}\n {% endfor -%}\n {% else -%}\n {{ content }}\n {% endif -%}\n {% endmacro -%}\n\n{% macro render_tool_calls(tool_calls) -%}\n {% for tool_call in tool_calls -%}\n {% if tool_call.function is defined -%}\n {% set tool_call = tool_call.function -%}\n {% endif -%}\n {{ '<tool_call>\n' }}\n {{ '{\"name\": \"' + tool_call.name + '\"' }}\n {% if tool_call.arguments is defined -%}\n {{ ', \"arguments\": ' }}\n {%# OpenAI-compatible clients send `arguments` as a JSON string; pass it through -#}\n {% if tool_call.arguments is string -%}\n {{ tool_call.arguments }}\n {% else -%}\n {{ tool_call.arguments | tojson }}\n {% endif -%}\n {{ '\n</tool_call>' }}\n {% endfor -%}\n\n{% endmacro -%}\n\n{% set ns = namespace(last_user_index=-1) %}\n {% for m in messages %}\n {% if m.role == 'user' %}\n {% set ns.last_user_index = loop.index0 %}\n {% endif %}\n {% endfor %}\n\n{% for message in messages %}\n {% if message.role == 'system' or message.role == 'user' %}\n {{ '<|im_start|>' + message.role + '\n' }}\n {% set reasoning_content = \" \" %}\n {% if message.reasoning_content is defined and message.reasoning_content is not none %}\n {% set reasoning_content = message.reasoning_content %}\n {% elif message.reasoning is defined and message.reasoning is not none %}\n {% set reasoning_content = message.reasoning %}\n {% endif %}\n {% if reasoning_content %}\n {{ '<think>' + reasoning_content.strip() + '</think>' }}\n {% endif %}\n {{ render_content(message.content) }}\n\n{% if message.tool_calls is defined and message.tool_calls is not none %}\n {{ render_tool_calls(message.tool_calls) }}\n {% endif %}\n {{ '<|im_end|>\n' }}\n\n{% elif message.role == 'assistant' %}\n {{ '<|im_start|>assistant\n' }}\n\n{%# Extract reasoning and content -#\n {% set reasoning_content = \" \" %}\n {% set content = \" \" %}\n {% if message.content is defined and message.content is not none %}\n {% set content = render_content(message.content) %}\n {% endif %}\n {% if message.reasoning_content is defined and message.reasoning_content is not none %}\n {% set reasoning_content = message.reasoning_content %}\n {% elif message.reasoning is defined and message.reasoning is not none %}\n {% set reasoning_content = message.reasoning %}\n {% endif %}\n {% if '<think>' in content %}\n {% set reasoning_content = content.split('</think>')[0].split('<think>')[1] %}\n {% set content = content.split('</think>')[1].rstrip('\n') %}\n {% endif %}\n\n{%# Determine whether to render thinking -#\n {% set render_thinking = false %}\n {% if reasoning_content %}\n {% if show_thinking is defined and show_thinking %}\n {% set render_thinking = true %}\n {% elif not (show_thinking is defined and not show_thinking) %}\n {%# show_thinking undefined: show if last turn or tool-call turn after last user -#\n {% if loop.last or (message.tool_calls is defined and loop.index0 > ns.last_user_index) %}\n {% set render_thinking = true %}\n {% endif %}\n {% endif %}\n {{ '<think>' + reasoning_content.strip() + '</think>' }}\n {% else %}\n {{ '</think>' }}\n {% endif %}\n\n{{ content }}\n\n{% if message.tool_calls is defined and message.tool_calls is not none %}\n {{ render_tool_calls(message.tool_calls) }}\n {% endif %}\n {{ '<|im_end|>' + ('\n' if (not loop.last or add_generation_prompt) else \" \" ) }}\n\n{% elif message.role == 'tool' %}\n {{ '<|im_start|>tool\n' }}\n {{ '<tool_response>\n' }}\n {{ render_content(message.content) }}\n {{ '\n' }}\n {{ '</tool_response><|im_end|>\n' }}\n {% endif %}\n\n{% endfor %}\n\n{% if add_generation_prompt %}\n {{ '<|im_start|>assistant\n' }}\n\n{% if enable_thinking is defined and enable_thinking is true %}\n {{ '<think>' }}\n {% else %}\n {{ '</think>' }}\n {% endif %}\n\n{% endfor %}\n\n{% endmacro -%}\n\n{%}\n\n\", \"createdAt\": \"2026-07-28T10:33:49.000Z\", \"discussionsDisabled\": false, \"discussionsSorting\": \"recently-created\", \"downloads\": 17096, \"downloadsAllTime\": 156524, \"id\": \"skt/A.X-K2\", \"isLikedByUser\": false, \"availableInferenceProviders\":\n\n[], \"showHuggingChatEntry\": false, \"inference\": \"\", \"lastModified\": \"2026-09-01T06:57:43.000Z\", \"likes\": 107, \"pipeline_tag\": \"generation\", \"library_name\": \"transformers\", \"librariesOther\": [], \"trackDownloads\": true, \"mask_token\": \"<|mask|>\", \"model-index\": null, \"evalResults\": [{\"dataset\": {\"id\": \"ldavidrein/gpqa\", \"isBenchmark\": true, \"task_id\": \"diamond\", \"value\": 85.6, \"source\": {\"url\": \"https://huggingface.co/skt/A.X-K2/blob/fb412dcd22e9c5ac43b2967d6c5e8a7d0e667b0f/README.md#L134-L152\", \"name\": \"Model Card\", \"isExternal\": false, \"filename\": \"eval_results/gpqa.yaml\", \"verified\": false, \"rank\": 44, \"label\": \"Diamond\"}, {\"dataset\": {\"id\": \"MathArena/aime_2026\", \"isBenchmark\": true, \"task_id\": \"MathArena/aime_2026\", \"value\": 97.1, \"source\": {\"url\": \"https://huggingface.co/skt/A.X-K2/blob/fb412dcd22e9c5ac43b2967d6c5e8a7d0e667b0f/README.md#L134-L152\", \"name\": \"Model

```

```

Card", "isExternal": false, "filename": ".eval_results/aime_2026.yaml", "verified": false, "rank": 4, "label": "MathArena
Aime 2026"}, {"dataset": {"id": "cais/hle", "isBenchmark": false, "task_id": "hle", "value": 27.8, "source": {"url": "https://
huggingface.co/skt/A.X-K2/blob/fb412dcd22e9c5ac43b2967d6c5e8a7d0e667b0f/README.md#L134-
L152", "name": "Model Card", "isExternal": false}, "filename": ".eval_results/
hle.yaml", "verified": false, "rank": 41, "label": "Hle"}], "private": false, "repoType": "model", "gated": false, "tags":
["transformers", "safetensors", "axk2", "text-generation", "conversational", "en", "ko", "zh", "ja", "es", "dataset:nvidia/
Nemotron-CC-v2.1", "dataset:nvidia/Nemotron-Pretraining-Code-v2", "dataset:HuggingFaceFW/
fineweb-2", "arxiv:2608.30181", "license:apache-2.0", "eval-
results", "endpoints_compatible", "fp8", "region:us"], "tag_objs": [{"id": "text-generation", "label": "Text
Generation", "type": "pipeline_tag", "subType": "nlp"}, {"id": "transformers", "label": "Transformers", "type": "library"},
{"id": "safetensors", "label": "Safetensors", "type": "library"}, {"id": "dataset:nvidia/Nemotron-CC-v2.1", "label": "nvidia/
Nemotron-CC-v2.1", "type": "dataset", "extra": {"disabled": false}}, {"id": "dataset:nvidia/Nemotron-Pretraining-Code-
v2", "label": "nvidia/Nemotron-Pretraining-Code-v2", "type": "dataset", "extra": {"disabled": false}},
{"id": "dataset:HuggingFaceFW/fineweb-2", "label": "HuggingFaceFW/fineweb-2", "type": "dataset", "extra":
{"disabled": false}}, {"id": "en", "label": "English", "type": "language"}, {"id": "ko", "label": "Korean", "type": "language"},
{"id": "zh", "label": "Chinese", "type": "language"}, {"id": "ja", "label": "Japanese", "type": "language"},
{"id": "es", "label": "Spanish", "type": "language"}, {"id": "axk2", "label": "axk2", "type": "other", "clickable": true},
{"id": "conversational", "label": "conversational", "type": "other", "clickable": true}, {"id": "eval-results", "label": "Eval
Results", "type": "other", "clickable": true}, {"id": "endpoints_compatible", "label": "Inference
Endpoints", "type": "other", "clickable": true}, {"id": "fp8", "label": "fp8", "type": "other", "clickable": true},
{"id": "arxiv:2608.30181", "label": "arxiv:2608.30181", "type": "arxiv", "extra": {"paperTitle": "A.X K2 Technical Report"}},
{"id": "license:apache-2.0", "label": "apache-2.0", "type": "license"}, {"type": "region", "label": "🌐 Region:
US", "id": "region:us"}], "transformersInfo": {"auto_model": "AutoModelForCausalLM", "pipeline_tag": "text-
generation", "processor": "AutoTokenizer"}, "widgetData": [{"text": "Hi, what can you help me with?"}, {"text": "What is
84 * 3 / 2?"}, {"text": "Tell me an interesting fact about the universe!"}, {"text": "Explain quantum computing in simple
terms."}], "safetensors": {"parameters":
{"F32": 15360, "BF16": 2515671296, "F8_E4M3": 689174544384, "total": 691690231040, "sharded": true, "totalFileSize": 691690231040},
"Evaluation results Idavidrein/gpqa · Diamond View evaluation results leaderboard 85.6

MathArena/aime_2026 · MathArena Aime 2026 View evaluation results leaderboard 97.1

cais/hle · Hle View evaluation results 27.8

```