

Tencent releases Hy4 preview, a 770-billion-parameter open model

Source copy kept by wheresthe.ai for the story "Tencent releases Hy4 preview, a 770-billion-parameter open model" (<https://wheresthe.ai/d/cn-tencent-hy4-preview-2026-08/>).

Source: <https://huggingface.co/tencent/Hy4-preview>

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

SHA-256 of the saved text: 4c64465e2fdcd08b0448e3b8eb022d101df2f858236f1a23e0b53eb4efeac82e

```
","eos_token": "< | hy_end:opensource | >","pad_token": "< | hy_pad:opensource | >"},"chat_template_jinja": "{#-
----- special token variables ----- #-}\n{%- set HYTK = ':opensource' %}\n{%- set hy_start_token = '< |
hy_start{} | >'.format(HYTK) %}\n{%- set hy_middle_token = '< | hy_middle{} | >'.format(HYTK) %}\n{%- set
hy_end_token = '< | hy_end{} | >'.format(HYTK) %}\n{%- set think_begin_token = '<think{}>'.format(HYTK)
%}\n{%- set think_end_token = '</think{}>'.format(HYTK) %}\n{%- set toolcalls_begin_token = '<tool_calls{}
>'.format(HYTK) %}\n{%- set toolcalls_end_token = '</tool_calls{}>'.format(HYTK) %}\n{%- set
toolcall_begin_token = '<tool_call{}>'.format(HYTK) %}\n{%- set toolcall_end_token = '</tool_call{}>'.format(HYTK)
%}\n{%- set argkey_begin_token = '<arg_key{}>'.format(HYTK) %}\n{%- set argkey_end_token = '</arg_key{}
>'.format(HYTK) %}\n{%- set argvalue_begin_token = '<arg_value{}>'.format(HYTK) %}\n{%- set
argvalue_end_token = '</arg_value{}>'.format(HYTK) %}\n{%- set toolresponse_begin_token = '<tool_response{}
>'.format(HYTK) %}\n{%- set toolresponse_end_token = '</tool_response{}>'.format(HYTK) %}\n{%- set
reasoning_mode_token = '< | reasoning_mode{} | >'.format(HYTK) %}\n\n#{# ----- hyperparameters
variables ----- #-}\n{%- if not add_generation_prompt is defined %}\n {%- set add_generation_prompt = false
%}\n{%- endif %}\n{%- if not preserved_thinking is defined %}\n {%- if not tools %}\n {%- set preserved_thinking =
false %}\n {%- else %}\n {%- set preserved_thinking = true %}\n {%- endif %}\n{%- endif %}\n\n{%- if not
reasoning_effort is defined %}\n {%- set reasoning_effort = 'high' %}\n{%- elif reasoning_effort not in ['high',
'no_think'] %}\n {%- if reasoning_effort is none %}\n {{- raise_exception('reasoning_effort error : None, should be
no_think/high') }}\n {%- else %}\n {{- raise_exception('reasoning_effort error : ' + reasoning_effort + ', should be
no_think/high') }}\n {%- endif %}\n{%- endif %}\n\n{%- if fallback_strategy is defined and fallback_strategy ==
'reasoning_toolcall_retry' %}\n {%- set reasoning_effort = 'high' %}\n {%- set add_generation_prompt = false
%}\n{%- endif %}\n{%- if not raw_last_assistant is defined %}\n {%- set raw_last_assistant = false %}\n{%- endif
%}\n\n{%- macro tool_to_json(tool) -%}\n {%- set ns_tool = namespace(first=true) -%}\n {{- ' ' -}}\n {%- for k, v in
tool.items() -%}\n {%- if k != 'defer_loading' and k != 'strict' -%}\n {%- if not ns_tool.first -%}\n {{- ' ' -}}\n {%- endif -%}\n
{%- set ns_tool.first = false -%}\n {{- '\n' ~ k ~ ':' ~ (v | tojson(ensure_ascii=False)) -}}\n {%- endif -%}\n {%- endfor
-%}\n {{- ' ' -}}\n{%- endmacro -%}\n\n{%- macro render_content(content) -%}\n {%- if content is string -%}\n {{-
content -}}\n {%- elif content is iterable and content is not mapping -%}\n {%- for item in content -%}\n {%- if item is
mapping and item.type == 'text' -%}\n {{- item.text -}}\n {%- elif item is string -%}\n {{- item -}}\n {%- else -%}\n {{-
item | string -}}\n {%- endif -%}\n {%- endfor -%}\n {%- elif content is none -%}\n {{- ' ' -}}\n {%- else -%}\n {{- content
| string -}}\n {%- endif -%}\n{%- endmacro -%}\n\n{%- macro render_tools_prompt() -%}\n {{- '# Tools\n\nYou may
call one or more functions to assist with the user query.' -}}\n {{- '\n\nYou are provided with function signatures
within <tools></tools> XML tags:' -}}\n {{- '\n<tools>\n' -}}\n {%- set tool_ns = namespace(first=true) -%}\n {%- for
tool in tools -%}\n {%- set t = tool['function'] if tool is mapping and 'function' in tool else tool -%}\n {%- if
t.defer_loading is not defined or not t.defer_loading -%}\n {%- if not tool_ns.first -%}\n {{- '\n' -}}\n {%- endif -%}\n
{%- set tool_ns.first = false -%}\n {{- tool_to_json(t) -}}\n {%- endif -%}\n {%- endfor -%}\n {{- '\n</tools>\n\n' -}}\n {{-
'For function call returns, you should first print ' ~ toolcalls_begin_token -}}\n {{- '\nFor each function call, you
should return object like:\n' -}}\n {{- toolcall_begin_token ~ '{function-name}' -}}\n {{- argkey_begin_token ~ '{arg-
key-1}' ~ argkey_end_token -}}\n {{- argvalue_begin_token ~ '{arg-value-1}' ~ argvalue_end_token -}}\n {{-
argkey_begin_token ~ '{arg-key-2}' ~ argkey_end_token -}}\n {{- argvalue_begin_token ~ '{arg-value-2}' ~
argvalue_end_token -}}\n {{- '...' -}}\n {{- toolcall_end_token -}}\n {{- '\nAt the end of function call returns, you
should print ' ~ toolcalls_end_token -}}\n{%- endmacro -%}\n\n{%- macro render_tool_response(message) -%}\n
{%- set content = message['content'] -%}\n {%- if content is string -%}\n {{- toolresponse_begin_token ~ content ~
toolresponse_end_token -}}\n {%- elif content is iterable and content is not mapping and content and content[0] is
mapping and content[0].type == 'tool_reference' -%}\n {{- toolresponse_begin_token -}}\n {{- '<tools>\n' -}}\n {%-
for tr in content -%}\n {%- for tool in tools -%}\n {%- set t = tool['function'] if tool is mapping and 'function' in tool
else tool -%}\n {%- if t.name == tr.name -%}\n {{- tool_to_json(t) ~ '\n' -}}\n {%- endif -%}\n {%- endfor -%}\n
{%- endfor -%}\n {{- '</tools>' -}}\n {{- toolresponse_end_token -}}\n {%- elif content is iterable and content is not
mapping and content and content[0] is mapping and content[0].output is defined -%}\n {%- for tr in content -%}\n
```

```

{{- toolresponse_begin_token ~ tr.output ~ toolresponse_end_token -}}\n {%- endfor -%}\n {%- else -%}\n {{-
toolresponse_begin_token ~ render_content(content) ~ toolresponse_end_token -}}\n {%- endif -%}\n{%-
endmacro -%}\n\n{%- consecutive tool messages are clustered into a single tool message whose content is the list
of the original tool messages; the result is written to merged_ns.messages (a macro cannot return a list) -#}\n{%-
set merged_ns = namespace(messages=[]) %}\n{%- macro merge_tool_responses(messages) -%}\n {%- set
merge_ns = namespace(result=[], cluster=[]) -%}\n {%- for message in messages -%}\n {%- if message['role'] ==
'tool' -%}\n {%- set merge_ns.cluster = merge_ns.cluster + [message] -%}\n {%- else -%}\n {%- if merge_ns.cluster
-%}\n {%- set merge_ns.result = merge_ns.result + [{'role': 'tool', 'content': merge_ns.cluster}] -%}\n {%- set
merge_ns.cluster = [] -%}\n {%- endif -%}\n {%- set merge_ns.result = merge_ns.result + [message] -%}\n {%-
endif -%}\n {%- endfor -%}\n {%- if merge_ns.cluster -%}\n {%- set merge_ns.result = merge_ns.result + [{'role':
'tool', 'content': merge_ns.cluster}] -%}\n {%- endif -%}\n {%- set merged_ns.messages = merge_ns.result -
%}\n{%- endmacro -%}\n\n{%- render one tool cluster, reordered to follow the tool_call order of the preceding
assistant turn; any missing / ambiguous id falls back to the original tool message order for the whole cluster -
#}\n{%- macro render_tool_responses(tool_messages, tool_call_ids) -%}\n {%- set order_ns =
namespace(ordered=[], matched=true) -%}\n {%- if tool_call_ids and tool_call_ids | length == tool_messages |
length -%}\n {%- for tool_call_id in tool_call_ids -%}\n {%- set hit_ns = namespace(count=0, message=None) -%}\n
{%- for tool_message in tool_messages -%}\n {%- set message_id = tool_message['tool_call_id'] if
tool_message['tool_call_id'] is defined else tool_message['id'] -%}\n {%- if message_id is defined and message_id
== tool_call_id -%}\n {%- set hit_ns.count = hit_ns.count + 1 -%}\n {%- set hit_ns.message = tool_message -%}\n
{%- endif -%}\n {%- endfor -%}\n {%- if hit_ns.count == 1 -%}\n {%- set order_ns.ordered = order_ns.ordered +
[hit_ns.message] -%}\n {%- else -%}\n {%- set order_ns.matched = false -%}\n {%- endif -%}\n {%- endfor -%}\n
{%- else -%}\n {%- set order_ns.matched = false -%}\n {%- endif -%}\n {%- for tool_message in (order_ns.ordered
if order_ns.matched else tool_messages) -%}\n {{- render_tool_response(tool_message) -}}\n {%- endfor -%}\n{%-
endmacro -%}\n\n{%- set _ = merge_tool_responses(messages) %}\n{%- set normed_messages =
merged_ns.messages %}\n\n{%- set ns = namespace(last_user_index=-1, has_leading_system=false) %}\n{%- if
normed_messages and normed_messages[0].role == 'system' %}\n {%- set ns.has_leading_system = true
%}\n{%- endif %}\n{%- for message in normed_messages %}\n {%- if message['role'] == 'user' %}\n {%- set
ns.last_user_index = loop.index0 %}\n {%- endif %}\n{%- endfor %}\n\n{%- no leading system: synthesize one for
tools and/or reasoning_mode -#}\n{%- if not ns.has_leading_system %}\n {{- hy_start_token ~ 'system' ~
hy_middle_token -}}\n {%- if tools %}\n {{- render_tools_prompt() -}}\n {%- endif %}\n {{- reasoning_mode_token ~
'reasoning_effort:' ~ reasoning_effort -}}\n {{- hy_end_token -}}\n{%- endif %}\n\n{%- set last_ns =
namespace(last_is_assistant=false) %}\n{%- set prev_ns = namespace(tool_call_ids=[]) %}\n{%- for message in
normed_messages %}\n {%- if message['role'] == 'system' %}\n {{- hy_start_token ~ 'system' ~ hy_middle_token
-}}\n {%- tools / reasoning_mode only attach to the leading system (messages[0]) -#}\n {%- if loop.first %}\n {%- if
tools %}\n {{- render_tools_prompt() -}}\n {%- endif %}\n {%- set content = render_content(message['content']) -
%}\n {%- if tools and content -%}\n {{- '\n\n' -}}\n {%- endif -%}\n {{- content -}}\n {{- reasoning_mode_token ~
'reasoning_effort:' ~ reasoning_effort -}}\n {%- else %}\n {{- render_content(message['content']) -}}\n {%- endif
%}\n {{- hy_end_token -}}\n {%- elif message['role'] == 'user' %}\n {{- hy_start_token ~ 'user' ~ hy_middle_token
-}}\n {{- render_content(message['content']) -}}\n {{- hy_end_token -}}\n {%- elif message['role'] == 'assistant' %}\n
{%- set content_body = render_content(message['content']) -%}\n {%- 'reasoning' takes precedence, fall back to
'reasoning_content' when it is empty -#}\n {%- set reasoning_text = " " %}\n {%- if message['reasoning'] is defined
and message['reasoning'] is string and message['reasoning'] %}\n {%- set reasoning_text = message['reasoning']
%}\n {%- elif message['reasoning_content'] is defined and message['reasoning_content'] is string and
message['reasoning_content'] %}\n {%- set reasoning_text = message['reasoning_content'] %}\n {%- endif %}\n
{%- no_think overrides preserved_thinking: history is always emitted with empty think tags -#}\n {%- if
reasoning_effort != 'no_think' and (preserved_thinking or loop.index0 > ns.last_user_index) and reasoning_text
%}\n {%- set content = think_begin_token ~ reasoning_text ~ think_end_token ~ content_body %}\n {%- else %}\n
{%- set content = think_begin_token ~ think_end_token ~ content_body %}\n {%- endif %}\n {{- hy_start_token ~
'assistant' ~ hy_middle_token -}}\n {%- if message['tool_calls'] is defined and message['tool_calls'] %}\n {%-
remember the tool_call ids of this turn to order the following tool responses; an incomplete id set means 'do not
reorder' -#}\n {%- set ids_ns = namespace(ids=[], complete=true) %}\n {%- for tool in message['tool_calls'] %}\n
{%- set tool_call_id = tool['id'] if tool['id'] is defined and tool['id'] else tool['tool_call_id'] %}\n {%- if tool_call_id is
defined and tool_call_id %}\n {%- set ids_ns.ids = ids_ns.ids + [tool_call_id] %}\n {%- else %}\n {%- set
ids_ns.complete = false %}\n {%- endif %}\n {%- endfor %}\n {%- set prev_ns.tool_call_ids = ids_ns.ids if
ids_ns.complete else [] %}\n {{- content -}}\n {{- toolcalls_begin_token -}}\n {%- for tool in message['tool_calls'] -
%}\n {%- set func = tool['function'] if tool is mapping and 'function' in tool else tool -%}\n {%- set arguments =

```

```

func['arguments'] -%}\n {{- toolcall_begin_token ~ func['name'] -}}\n {{- for key, value in arguments.items() -}}\n
{{- argkey_begin_token ~ key ~ argkey_end_token -}}\n {{- if value is not string -}}\n {{- set value = value |
tojson(ensure_ascii=False) -}}\n {{- endif -}}\n {{- argvalue_begin_token ~ value ~ argvalue_end_token -}}\n
{{- endfor -}}\n {{- toolcall_end_token -}}\n {{- endfor -}}\n {{- toolcalls_end_token -}}\n {{- else %}}\n {{- if
loop.last and raw_last_assistant %}}\n {{- content_body -}}\n {{- else %}}\n {{- content -}}\n {{- endif %}}\n {{- endif
%}}\n {{- continuation / prefill: last assistant is still open, do not close with hy_end -#}}\n {{- if not (loop.last and
raw_last_assistant) %}}\n {{- hy_end_token -}}\n {{- endif %}}\n {{- elif message['role'] == 'tool' %}}\n {{-
hy_start_token ~ 'tool' ~ hy_middle_token -}}\n {{- render_tool_responses(message['content'],
prev_ns.tool_call_ids) -}}\n {{- hy_end_token -}}\n {{- else %}}\n {{- hy_start_token ~ message['role'] ~
hy_middle_token -}}\n {{- render_content(message['content']) -}}\n {{- hy_end_token -}}\n {{- endif %}}\n {{- if
message['role'] != 'assistant' or not (message['tool_calls'] is defined and message['tool_calls']) %}}\n {{- set
prev_ns.tool_call_ids = [] %}}\n {{- endif %}}\n {{- if loop.last and message['role'] == 'assistant' %}}\n {{- set
last_ns.last_is_assistant = true %}}\n {{- endif %}}\n {{- endfor %}}\n {{- if add_generation_prompt %}}\n {{- if not
last_ns.last_is_assistant %}}\n {{- if reasoning_effort == 'no_think' %}}\n {{- hy_start_token ~ 'assistant' ~
hy_middle_token ~ think_begin_token ~ think_end_token -}}\n {{- else %}}\n {{- hy_start_token ~ 'assistant' ~
hy_middle_token ~ think_begin_token -}}\n {{- endif %}}\n {{- endif %}}\n {{- endif %}}\n {{- endif %}}\n
%}}\n {{- created_at: "2026-08-27T08:52:40.000Z", discussions_disabled: false, discussions_sorting: "recently-
created", downloads: 22357, downloads_all_time: 36987, id: "tencent/Hy4-
preview", is_liked_by_user: false, available_inference_providers:
[{"provider": "novita", model_status: "live", provider_status: "live", provider_id: "tencent/hy4-
preview", task: "conversational", adapter_weights_path: "model-00001-of-00131.safetensors", features:
{"structured_output": true, tool_calling: true, is_cheapest_pricing_output: true, is_fastest_throughput: false, is_model_authc:
{"provider": "deepinfra", model_status: "live", provider_status: "live", provider_id: "tencent/Hy4-
preview", task: "conversational", adapter_weights_path: "model-00001-of-00131.safetensors", features:
{"structured_output": true, tool_calling: true, is_cheapest_pricing_output: false, is_fastest_throughput: true, is_model_authc:
generation, library_name: "transformers", libraries_other: [], track_downloads: true, model-
index": null, eval_results: [{"dataset": {"id": "Idavidrein/
gpqa", is_benchmark: true, task_id: "diamond"}, {"value": 92.3, "source": {"url": "https://huggingface.co/tencent/Hy4-
preview", "name": "Model Card", is_external: false, filename: ".eval_results/
gpqa.yaml", "verified": false, "pull_request": 3, "rank": 6, "label": "Diamond"}, {"dataset": {"id": "ScaleAI/SWE-
bench_Pro", is_benchmark: true, task_id: "SWE_Bench_Pro"}, {"value": 65.7, "source": {"url": "https://huggingface.co/
tencent/Hy4-preview", "name": "Model Card", is_external: false, filename: ".eval_results/swe-
bench_pro.yaml", "verified": false, "pull_request": 3, "rank": 4, "label": "SWE Bench Pro"}, {"dataset": {"id": "mercor/apex-
agents", is_benchmark: true, task_id: "apex-agents"}, {"value": 37.1, "source": {"url": "https://huggingface.co/tencent/
Hy4-preview", "name": "Model Card", is_external: false, filename: ".eval_results/apex-
agents.yaml", "verified": false, "pull_request": 12, "rank": 2, "label": "Apex Agents"}, {"dataset": {"id": "datacurve/deep-
swe", is_benchmark: true, task_id: "deep_swe"}, {"value": 64.3, "source": {"url": "https://huggingface.co/tencent/Hy4-
preview", "name": "Model Card", is_external: false, filename: ".eval_results/deep-
swe.yaml", "verified": false, "pull_request": 3, "rank": 8, "label": "Deep Swe"}, {"dataset": {"id": "harborframework/terminal-
bench-2.1", is_benchmark: true, task_id: "terminalbench_2_1"}, {"value": 85.4, "source": {"url": "https://huggingface.co/
tencent/Hy4-preview", "name": "Model Card", is_external: false, filename: ".eval_results/terminal-
bench-2.1.yaml", "verified": false, "pull_request": 12, "rank": 8, "label": "Terminalbench 2 1"}, {"dataset":
{"id": "IntelligenceLab/Long-Horizon-Terminal-
Bench", is_benchmark: true, task_id: "lhtb2_solved"}, {"value": 15, "source": {"url": "https://github.com/zli12321/LHTB/
tree/lhtb-2.0#lhtb-20-results-october-2026", "name": "LHTB 2.0 results", is_external: true, filename: ".eval_results/
lhtb2_solved.yaml", "verified": false, "pull_request": 14, "rank": 2, "label": "Lhtb2 Solved", "notes": "15/77 tasks solved at
reward >= 0.95 (pass@0.95); mean reward x100 = 53.2; LHTB 2.0 (77 tasks), terminus-2, 3-hour budget per task,
official LHTB Harbor harness"}, {"dataset": {"id": "hkust-nlp/
Toolathlon", is_benchmark: false, task_id: "toolathlon_verified"}, {"value": 74.1, "source": {"url": "https://huggingface.co/
tencent/Hy4-preview", "name": "tencent/Hy4-preview model card", is_external: false, filename: ".eval_results/
tencent__Hy4-preview.yaml", "verified": false, "pull_request": 13, "rank": 5, "label": "Toolathlon Verified"}, {"dataset":
{"id": "cais/hle", is_benchmark: false, task_id: "hle"}, {"value": 43.4, "source": {"url": "https://huggingface.co/tencent/
Hy4-preview", "name": "Model Card", is_external: false, filename: ".eval_results/
hle.yaml", "verified": false, "pull_request": 12, "label": "Hle", "notes": "Text-only, no tools."}, {"dataset": {"id": "cais/
hle", is_benchmark: false, task_id: "hle"}, {"value": 55.4, "source": {"url": "https://huggingface.co/tencent/Hy4-
preview", "name": "Model Card", is_external: false, filename: ".eval_results/

```

```
hle.yaml", "verified": false, "pullRequest": 12, "rank": 6, "label": "Hle", "notes": "Text-only, with tools."}, {"dataset":
{"id": "benchflow/skillsbench", "isBenchmark": true, "task_id": "skillsbench_v1_1", "value": 62.9, "source": {"url": "https://
huggingface.co/tencent/Hy4-preview", "name": "Model Card", "isExternal": false}, "filename": ".eval_results/
skillsbench.yaml", "verified": false, "pullRequest": 3, "rank": 2, "label": "Skillsbench V1 1"}, {"dataset": {"id": "SWE-bench/
SWE-
bench_Multilingual", "isBenchmark": true, "task_id": "swe_bench_multilingual_%_resolved"}, {"value": 82.9, "source":
{"url": "https://huggingface.co/tencent/Hy4-preview", "name": "Model
Card", "isExternal": false}, "filename": ".eval_results/swe-
bench_multilingual.yaml", "verified": false, "pullRequest": 3, "rank": 2, "label": "Swe Bench Multilingual
Resolved"}], "private": false, "repoType": "model", "gated": false, "tags": [{"transformers", "safetensors", "hy_v4", "text-
generation", "hunyuan", "hy4", "moe", "conversational", "arxiv:2512.02556", "arxiv:2603.12201", "license:apache-2.0", "eval
results", "endpoints_compatible", "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": "hy_v4", "label": "hy_v4", "type": "other", "clickable": true},
{"id": "hunyuan", "label": "hunyuan", "type": "other", "clickable": true},
{"id": "hy4", "label": "hy4", "type": "other", "clickable": true}, {"id": "moe", "label": "Mixture of
Experts", "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": "arxiv:2512.02556", "label": "arxiv:2512.02556", "type": "arxiv", "extra":
{"paperTitle": "DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models"}],
{"id": "arxiv:2603.12201", "label": "arxiv:2603.12201", "type": "arxiv", "extra": {"paperTitle": "IndexCache: Accelerating
Sparse Attention via Cross-Layer Index Reuse"}}, {"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":
{"BF16": 779930197312, "F32": 30795421}, {"total": 779960992733, "sharded": true, "totalFileSize": 1559983985227}, {"hasE
{"closed": 1, "open": 13, "total": 14}, {"query": {}, "inferenceContextData": {"billableEntities": [], "entityName2Providers":
{}}, {"defaultProviders":
[{"isOriginalProvider": false, "name": "novita", "enabled": true, "position": 1, "isReleased": true, "accuratePricing": true},
{"isOriginalProvider": false, "name": "deepinfra", "enabled": true, "position": 15, "isReleased": true, "accuratePricing": true}], '
[]}]>
```

[中文](#) | [English](#)

[🌐 Official Website](#) | [💬 GitHub](#)

[Table of Contents](#)

[Model Introduction](#)

[A New Flagship Generation](#)

[Built for Productivity](#)

[Benchmark Appendix](#)

[Known Limitations](#)

[News](#)

[Model Links](#)

[Quickstart](#)

[Deployment vLLM](#)

[SGLang](#)

[Finetuning](#)

Quantization

License

Contact Us

Model Introduction

Hy4 preview is a new-generation Mixture-of-Experts (MoE) flagship model developed by the Tencent Hy Team. The model comprises 770B total parameters, of which 49B are activated per token. The backbone consists of 78 layers, where the first layer uses a standard dense FFN and the remaining 77 layers replace it with MoE, each containing 256 routed experts and 1 shared expert; every token activates the top-8 routed experts along with the shared expert. In addition to the backbone, 1 native MTP layer (10B total parameters, 0.7B activated) is built in for speculative decoding.

On the architecture side, inspired by DeepSeek and GLM, the attention module employs Gated DeepSeek Sparse Attention (Gated DSA) with IndexCache for cross-layer sparse index reuse. The residual pathway uses iHC (identity Hyper-Connections) to expand inter-layer information flow.

Model Specifications

The table below lists backbone parameters only, excluding the MTP layer.

Property Value

Architecture Mixture-of-Experts (MoE)

Total Parameters 770B

Activated Parameters 49B

Layers 78

Hidden Size 6144

Attention Type Gated DSA

Attention Heads 64

Query Compression Dimension 2048

Key-Value Compression Dimension 512

Indexer Heads / Head Dimension 32 / 128

Indexer top-k 2048

Residual Streams 4

Routed Experts 256

Shared Experts 1

Activated Routed Experts per Token 8

MoE Intermediate Size 2048

FFN Intermediate Size 18432

Context Length 1M

Vocabulary Size 120832

A New Flagship Generation

We scaled Hy4 preview on three fronts: model size, context length, and training data. Stronger pre-training and a substantially larger post-training run compound into another step change in capability — the largest generation-over-generation gain we've measured, and enough to put Hy4 preview at the open-source frontier.

Built for Productivity

We partnered with top experts inside Tencent — such as software engineers, game developers, finance analysts, and security experts — and built training data around the work they ship. The result is a model that gets meaningfully further on the tasks these teams run every day:

Software engineering : Better at understanding, planning, debugging, and verifying long-horizon development tasks, with further gains in the visual taste and interaction quality of front-end work.

Office and analysis : Takes messy context spread across many files and converts it into shareable artifacts — documents, spreadsheets, and presentations — handling data analysis, equations, and financial models with greater precision.

Game development : Turns a single prompt into a playable prototype and works fluently with game engines, so developers can keep refining complex projects over multiple turns.

Scientific research : Stronger understanding, reasoning, and problem-solving on hard research questions, with solid progress across AI research, molecular dynamics, condensed matter physics, and pure mathematics.

We also continue to co-design Hy4 preview with Tencent products like CodeBuddy and WorkBuddy, so that gains in the model show up in the work people actually do with it. To check that, we ran a blind side-by-side evaluation: 163 internal experts rated model outputs on 203 engineering tasks. Hy4 preview came out slightly ahead of both GLM 5.3 (2.99 vs. 2.92 average, 46.8% wins / 12.8% ties / 40.4% losses) and Kimi K3 (2.99 vs. 2.94, 51.2% wins / 7.9% ties / 40.9% losses).

Benchmark Appendix

Known Limitations

This is an early version of Hy4. There is real headroom left in both pre-training and post-training, and we are shipping with known issues — among them, spending longer than necessary reasoning through complex tasks, and a tendency to over-verify its own work. We'll keep iterating quickly on these. As with Hy3 preview, we would rather ship early and hear what breaks — that's what made Hy3 substantially better, and it's how we will get Hy4 right. We will also keep collaborating closely with Tencent's products and in-house experts to push the boundaries of model intelligence while making it more abundant and affordable.

News

🔥 We open-source Hy4 preview and Hy4 preview-FP8 model weights on Hugging Face , ModelScope , GitHub , and CNB .

Model Links

Model Name	Description	Hugging Face	ModelScope	GitHub	CNB
------------	-------------	--------------	------------	--------	-----

Hy4 preview	Instruct model	Model	Model	Model	Model
-------------	----------------	-------	-------	-------	-------

Hy4 preview-FP8	FP8 quantized instruct model	Model	Model	Model	Model
-----------------	------------------------------	-------	-------	-------	-------

Quickstart

Deploy Hy4 preview with vLLM or SGLang first, then call the OpenAI-compatible API:

```
from openai import OpenAI
```

```
client = OpenAI(base_url="http://127.0.0.1:8000/v1", api_key="EMPTY")
```

```
response = client.chat.completions.create(model="hy4-preview", messages=[{"role": "user", "content": "Hello! Can you briefly introduce yourself?"}], temperature=0.9, top_p=1.0) print(response.choices[0].message.content)
```

Recommended parameters : temperature=0.9 , top_p=1.0 .

Reasoning mode : Defaults to "high" (deep chain-of-thought), which suits complex tasks such as math, coding, and reasoning. For direct responses, pass extra_body={"chat_template_kwargs": {"reasoning_effort": "no_think"}}

See the Deployment section below for how to start the API server.

Deployment

For production serving, we recommend using vLLM or SGLang . Please refer to the recipes:

Hy4-Preview vLLM Recipe

Hy4-Preview SGLang Cookbook

vLLM

Use official prebuilt image vllm/vllm-openai:hy4-preview :

```
docker run --gpus all \ -p 8000:8000 \ --ipc=host \ -v ~/.cache/huggingface:/root/.cache/huggingface \ vllm/vllm-openai:hy4-preview tencent/Hy4-preview-FP8 \ --tensor-parallel-size 8 \ --speculative-config '{"num_speculative_tokens":3,"method":"mtp"}' \ --attention-backend FLASHMLA_SPARSE \ --tool-call-parser hy_v4 \ --reasoning-parser hy_v4 \ --enable-auto-tool-choice \ --port 8000 \ --served-model-name hy4-preview
```

SGLang

Use the official prebuilt image lmsysorg/sglang:hy4-preview (multi-arch, x86 and Arm):

```
docker pull lmsysorg/sglang:hy4-preview
```

```
docker run --gpus all --ipc=host -p 8000:8000 lmsysorg/sglang:hy4-preview \ python3 -m sglang.launch_server \ --model tencent/Hy4-preview-FP8 \ --tp-size 8 \ --reasoning-parser auto \ --tool-call-parser auto \ --speculative-algorithm NEXTN \ --speculative-num-steps 3 \ --speculative-eagle-topk 1 \ --speculative-num-draft-tokens 4 \ --port 8000 \ --served-model-name hy4-preview
```

Finetuning

Hy4 preview provides a complete model finetuning pipeline. For detailed documentation, please refer to: [Finetuning Guide](#)

Quantization

We provide AngelSlim , a more accessible, comprehensive, and efficient toolkit for large model compression. AngelSlim supports a comprehensive suite of compression tools for large-scale multimodal models, including common quantization algorithms, low-bit quantization, and speculative sampling.

License

Hy4 preview is released under the Apache License 2.0 . See LICENSE for details.

Contact Us

If you have any questions or suggestions, feel free to reach out to our R&D and product teams via email:

✉ hunyuan_opensource@tencent.com

Hy4 preview is developed by the Tencent Hy Team.

Downloads last month 22,357

```
"eos_token": "< | hy_end:opensource | >", "pad_token": "< | hy_pad:opensource | >", "chat_template_jinja": "{#-
----- special token variables ----- #}\n{% set HYTK = ':opensource' %}\n{% set hy_start_token = '< |
hy_start{} | >'.format(HYTK) %}\n{% set hy_middle_token = '< | hy_middle{} | >'.format(HYTK) %}\n{% set
hy_end_token = '< | hy_end{} | >'.format(HYTK) %}\n{% set think_begin_token = '<think{}>'.format(HYTK)
%}\n{% set think_end_token = '</think{}>'.format(HYTK) %}\n{% set toolcalls_begin_token = '<tool_calls{}
>'.format(HYTK) %}\n{% set toolcalls_end_token = '</tool_calls{}>'.format(HYTK) %}\n{% set
toolcall_begin_token = '<tool_call{}>'.format(HYTK) %}\n{% set toolcall_end_token = '</tool_call{}>'.format(HYTK)
%}\n{% set argkey_begin_token = '<arg_key{}>'.format(HYTK) %}\n{% set argkey_end_token = '</arg_key{}
>'.format(HYTK) %}\n{% set argvalue_begin_token = '<arg_value{}>'.format(HYTK) %}\n{% set
argvalue_end_token = '</arg_value{}>'.format(HYTK) %}\n{% set toolresponse_begin_token = '<tool_response{}
>'.format(HYTK) %}\n{% set toolresponse_end_token = '</tool_response{}>'.format(HYTK) %}\n{% set
reasoning_mode_token = '< | reasoning_mode{} | >'.format(HYTK) %}\n\n\n#- ----- hyperparameters
variables ----- #}\n{% if not add_generation_prompt is defined %}\n\n{% set add_generation_prompt = false
```

```

%}\n{%\n - endif %}\n{%\n - if not preserved_thinking is defined %}\n{%\n - if not tools %}\n{%\n - set preserved_thinking =
false %}\n{%\n - else %}\n{%\n - set preserved_thinking = true %}\n{%\n - endif %}\n{%\n - endif %}\n{%\n - if not
reasoning_effort is defined %}\n{%\n - set reasoning_effort = 'high' %}\n{%\n - elif reasoning_effort not in ['high',
'no_think'] %}\n{%\n - if reasoning_effort is none %}\n{%\n - raise_exception('reasoning_effort error : None, should be
no_think/high') %}\n{%\n - else %}\n{%\n - raise_exception('reasoning_effort error : ' + reasoning_effort + ', should be
no_think/high') %}\n{%\n - endif %}\n{%\n - endif %}\n{%\n - if fallback_strategy is defined and fallback_strategy ==
'reasoning_toolcall_retry' %}\n{%\n - set reasoning_effort = 'high' %}\n{%\n - set add_generation_prompt = false
%}\n{%\n - endif %}\n{%\n - if not raw_last_assistant is defined %}\n{%\n - set raw_last_assistant = false %}\n{%\n - endif
%}\n{%\n - macro tool_to_json(tool) %}\n{%\n - set ns_tool = namespace(first=true) %}\n{%\n - {{- ' ' -}}\n{%\n - for k, v in
tool.items() %}\n{%\n - if k != 'defer_loading' and k != 'strict' %}\n{%\n - if not ns_tool.first %}\n{%\n - {{- ' ' -}}\n{%\n - endif %}\n{%\n - set ns_tool.first = false %}\n{%\n - {{- '\n' ~ k ~ '\n': ' ~ (v | tojson(ensure_ascii=False)) -}}\n{%\n - endif %}\n{%\n - endfor
%}\n{%\n - {{- ' ' -}}\n{%\n - endmacro %}\n{%\n - macro render_content(content) %}\n{%\n - if content is string %}\n{%\n - {{-
content -}}\n{%\n - elif content is iterable and content is not mapping %}\n{%\n - for item in content %}\n{%\n - if item is
mapping and item.type == 'text' %}\n{%\n - {{- item.text -}}\n{%\n - elif item is string %}\n{%\n - {{- item -}}\n{%\n - else %}\n{%\n - {{-
item | string -}}\n{%\n - endif %}\n{%\n - endfor %}\n{%\n - elif content is none %}\n{%\n - {{- ' ' -}}\n{%\n - else %}\n{%\n - {{- content
| string -}}\n{%\n - endif %}\n{%\n - endmacro %}\n{%\n - macro render_tools_prompt() %}\n{%\n - {{- '# Tools\n\nYou may
call one or more functions to assist with the user query.' -}}\n{%\n - {{- '\n\nYou are provided with function signatures
within <tools></tools> XML tags:' -}}\n{%\n - {{- '\n<tools>\n\n' -}}\n{%\n - set tool_ns = namespace(first=true) %}\n{%\n - for
tool in tools %}\n{%\n - set t = tool['function'] if tool is mapping and 'function' in tool else tool %}\n{%\n - if
t.defer_loading is not defined or not t.defer_loading %}\n{%\n - if not tool_ns.first %}\n{%\n - {{- '\n' -}}\n{%\n - endif %}\n{%\n - set
tool_ns.first = false %}\n{%\n - {{- tool_to_json(t) -}}\n{%\n - endif %}\n{%\n - endfor %}\n{%\n - {{- '\n</tools>\n\n' -}}\n{%\n - {{-
'For function call returns, you should first print ' ~ toolcalls_begin_token -}}\n{%\n - {{- '\nFor each function call, you
should return object like:\n\n' -}}\n{%\n - {{- toolcall_begin_token ~ '{function-name}' -}}\n{%\n - {{- argkey_begin_token ~ '{arg-
key-1}' ~ argkey_end_token -}}\n{%\n - {{- argvalue_begin_token ~ '{arg-value-1}' ~ argvalue_end_token -}}\n{%\n - {{-
argkey_begin_token ~ '{arg-key-2}' ~ argkey_end_token -}}\n{%\n - {{- argvalue_begin_token ~ '{arg-value-2}' ~
argvalue_end_token -}}\n{%\n - {{- '...' -}}\n{%\n - {{- toolcall_end_token -}}\n{%\n - {{- '\nAt the end of function call returns, you
should print ' ~ toolcalls_end_token -}}\n{%\n - endmacro %}\n{%\n - macro render_tool_response(message) %}\n{%\n - set
content = message['content'] %}\n{%\n - if content is string %}\n{%\n - {{- toolresponse_begin_token ~ content ~
toolresponse_end_token -}}\n{%\n - elif content is iterable and content is not mapping and content and content[0] is
mapping and content[0].type == 'tool_reference' %}\n{%\n - {{- toolresponse_begin_token -}}\n{%\n - {{- '<tools>\n\n' -}}\n{%\n - for
tr in content %}\n{%\n - for tool in tools %}\n{%\n - set t = tool['function'] if tool is mapping and 'function' in tool
else tool %}\n{%\n - if t.name == tr.name %}\n{%\n - {{- tool_to_json(t) ~ '\n' -}}\n{%\n - endif %}\n{%\n - endfor %}\n{%\n - endfor
%}\n{%\n - {{- '</tools>' -}}\n{%\n - {{- toolresponse_end_token -}}\n{%\n - elif content is iterable and content is not
mapping and content and content[0] is mapping and content[0].output is defined %}\n{%\n - for tr in content %}\n{%\n - {{-
toolresponse_begin_token ~ tr.output ~ toolresponse_end_token -}}\n{%\n - endfor %}\n{%\n - else %}\n{%\n - {{-
toolresponse_begin_token ~ render_content(content) ~ toolresponse_end_token -}}\n{%\n - endif %}\n{%\n - endmacro %}\n{%\n - {{-
#- consecutive tool messages are clustered into a single tool message whose content is the list
of the original tool messages; the result is written to merged_ns.messages (a macro cannot return a list) -#}}\n{%\n - set
merged_ns = namespace(messages=[]) %}\n{%\n - macro merge_tool_responses(messages) %}\n{%\n - set
merge_ns = namespace(result=[], cluster=[]) %}\n{%\n - for message in messages %}\n{%\n - if message['role'] ==
'tool' %}\n{%\n - set merge_ns.cluster = merge_ns.cluster + [message] %}\n{%\n - else %}\n{%\n - if merge_ns.cluster
%}\n{%\n - set merge_ns.result = merge_ns.result + [{'role': 'tool', 'content': merge_ns.cluster}] %}\n{%\n - set
merge_ns.cluster = [] %}\n{%\n - endif %}\n{%\n - set merge_ns.result = merge_ns.result + [message] %}\n{%\n - endif
%}\n{%\n - endfor %}\n{%\n - if merge_ns.cluster %}\n{%\n - set merge_ns.result = merge_ns.result + [{'role':
'tool', 'content': merge_ns.cluster}] %}\n{%\n - endif %}\n{%\n - set merged_ns.messages = merge_ns.result -
%}\n{%\n - endmacro %}\n{%\n - {{- #- render one tool cluster, reordered to follow the tool_call order of the preceding
assistant turn; any missing / ambiguous id falls back to the original tool message order for the whole cluster -
#}}\n{%\n - macro render_tool_responses(tool_messages, tool_call_ids) %}\n{%\n - set order_ns =
namespace(ordered=[], matched=true) %}\n{%\n - if tool_call_ids and tool_call_ids | length == tool_messages |
length %}\n{%\n - for tool_call_id in tool_call_ids %}\n{%\n - set hit_ns = namespace(count=0, message=none) %}\n{%\n - for
tool_message in tool_messages %}\n{%\n - set message_id = tool_message['tool_call_id'] if
tool_message['tool_call_id'] is defined else tool_message['id'] %}\n{%\n - if message_id is defined and message_id
== tool_call_id %}\n{%\n - set hit_ns.count = hit_ns.count + 1 %}\n{%\n - set hit_ns.message = tool_message %}\n{%\n -
endif %}\n{%\n - endfor %}\n{%\n - if hit_ns.count == 1 %}\n{%\n - set order_ns.ordered = order_ns.ordered +
[hit_ns.message] %}\n{%\n - else %}\n{%\n - set order_ns.matched = false %}\n{%\n - endif %}\n{%\n - endfor %}\n{%\n - else
%}\n{%\n - set order_ns.matched = false %}\n{%\n - endif %}\n{%\n - for tool_message in (order_ns.ordered

```

```

if order_ns.matched else tool_messages) %}\n {{- render_tool_response(tool_message) -}}\n {{%- endfor -%}}\n{{%-
endmacro -%}}\n\n{{%- set _ = merge_tool_responses(messages) %}}\n{{%- set normed_messages =
merged_ns.messages %}}\n\n{{%- set ns = namespace(last_user_index=-1, has_leading_system=false) %}}\n{{%- if
normed_messages and normed_messages[0].role == 'system' %}}\n{{%- set ns.has_leading_system = true
%}}\n{{%- endif %}}\n{{%- for message in normed_messages %}}\n{{%- if message['role'] == 'user' %}}\n{{%- set
ns.last_user_index = loop.index0 %}}\n{{%- endif %}}\n{{%- endfor %}}\n\n{{%- no leading system: synthesize one for
tools and/or reasoning_mode -# %}}\n{{%- if not ns.has_leading_system %}}\n{{- hy_start_token ~ 'system' ~
hy_middle_token -}}\n{{%- if tools %}}\n{{- render_tools_prompt() -}}\n{{%- endif %}}\n{{- reasoning_mode_token ~
'reasoning_effort:' ~ reasoning_effort -}}\n{{- hy_end_token -}}\n{{%- endif %}}\n\n{{%- set last_ns =
namespace(last_is_assistant=false) %}}\n{{%- set prev_ns = namespace(tool_call_ids=[]) %}}\n{{%- for message in
normed_messages %}}\n{{%- if message['role'] == 'system' %}}\n{{- hy_start_token ~ 'system' ~ hy_middle_token
-}}\n{{- #- tools / reasoning_mode only attach to the leading system (messages[0]) -# %}}\n{{%- if loop.first %}}\n{{%- if
tools %}}\n{{- render_tools_prompt() -}}\n{{%- endif %}}\n{{%- set content = render_content(message['content']) -
%}}\n{{%- if tools and content -%}}\n{{- '\n\n' -}}\n{{%- endif -%}}\n{{- content -}}\n{{- reasoning_mode_token ~
'reasoning_effort:' ~ reasoning_effort -}}\n{{%- else %}}\n{{- render_content(message['content']) -}}\n{{%- endif
%}}\n{{- hy_end_token -}}\n{{%- elif message['role'] == 'user' %}}\n{{- hy_start_token ~ 'user' ~ hy_middle_token
-}}\n{{- render_content(message['content']) -}}\n{{- hy_end_token -}}\n{{%- elif message['role'] == 'assistant' %}}\n
{{%- set content_body = render_content(message['content']) -%}}\n{{- #- 'reasoning' takes precedence, fall back to
'reasoning_content' when it is empty -# %}}\n{{%- set reasoning_text = " %}}\n{{%- if message['reasoning'] is defined
and message['reasoning'] is string and message['reasoning'] %}}\n{{%- set reasoning_text = message['reasoning']
%}}\n{{%- elif message['reasoning_content'] is defined and message['reasoning_content'] is string and
message['reasoning_content'] %}}\n{{%- set reasoning_text = message['reasoning_content'] %}}\n{{%- endif %}}\n
{{- #- no_think overrides preserved_thinking: history is always emitted with empty think tags -# %}}\n{{%- if
reasoning_effort != 'no_think' and (preserved_thinking or loop.index0 > ns.last_user_index) and reasoning_text
%}}\n{{%- set content = think_begin_token ~ reasoning_text ~ think_end_token ~ content_body %}}\n{{%- else %}}\n
{{%- set content = think_begin_token ~ think_end_token ~ content_body %}}\n{{%- endif %}}\n{{- hy_start_token ~
'assistant' ~ hy_middle_token -}}\n{{%- if message['tool_calls'] is defined and message['tool_calls'] %}}\n{{- #-
remember the tool_call ids of this turn to order the following tool responses; an incomplete id set means "do not
reorder" -# %}}\n{{%- set ids_ns = namespace(ids=[], complete=true) %}}\n{{%- for tool in message['tool_calls'] %}}\n
{{%- set tool_call_id = tool['id'] if tool['id'] is defined and tool['id'] else tool['tool_call_id'] %}}\n{{%- if tool_call_id is
defined and tool_call_id %}}\n{{%- set ids_ns.ids = ids_ns.ids + [tool_call_id] %}}\n{{%- else %}}\n{{%- set
ids_ns.complete = false %}}\n{{%- endif %}}\n{{%- endfor %}}\n{{%- set prev_ns.tool_call_ids = ids_ns.ids if
ids_ns.complete else [] %}}\n{{- content -}}\n{{- toolcalls_begin_token -}}\n{{%- for tool in message['tool_calls'] -
%}}\n{{%- set func = tool['function'] if tool is mapping and 'function' in tool else tool -%}}\n{{%- set arguments =
func['arguments'] -%}}\n{{- toolcall_begin_token ~ func['name'] -}}\n{{%- for key, value in arguments.items() -%}}\n
{{- argkey_begin_token ~ key ~ argkey_end_token -}}\n{{%- if value is not string -%}}\n{{%- set value = value |
tojson(ensure_ascii=False) -%}}\n{{%- endif -%}}\n{{- argvalue_begin_token ~ value ~ argvalue_end_token -}}\n
{{%- endfor -%}}\n{{- toolcall_end_token -}}\n{{%- endfor -%}}\n{{- toolcalls_end_token -}}\n{{%- else %}}\n{{%- if
loop.last and raw_last_assistant %}}\n{{- content_body -}}\n{{%- else %}}\n{{- content -}}\n{{%- endif %}}\n{{%- endif
%}}\n{{- #- continuation / prefill: last assistant is still open, do not close with hy_end -# %}}\n{{%- if not (loop.last and
raw_last_assistant) %}}\n{{- hy_end_token -}}\n{{%- endif %}}\n{{%- elif message['role'] == 'tool' %}}\n{{-
hy_start_token ~ 'tool' ~ hy_middle_token -}}\n{{- render_tool_responses(message['content'],
prev_ns.tool_call_ids) -}}\n{{- hy_end_token -}}\n{{%- else %}}\n{{- hy_start_token ~ message['role'] ~
hy_middle_token -}}\n{{- render_content(message['content']) -}}\n{{- hy_end_token -}}\n{{%- endif %}}\n{{%- if
message['role'] != 'assistant' or not (message['tool_calls'] is defined and message['tool_calls']) %}}\n{{%- set
prev_ns.tool_call_ids = [] %}}\n{{%- endif %}}\n{{%- if loop.last and message['role'] == 'assistant' %}}\n{{%- set
last_ns.last_is_assistant = true %}}\n{{%- endif %}}\n{{%- endfor %}}\n{{%- if add_generation_prompt %}}\n{{%- if not
last_ns.last_is_assistant %}}\n{{%- if reasoning_effort == 'no_think' %}}\n{{- hy_start_token ~ 'assistant' ~
hy_middle_token ~ think_begin_token ~ think_end_token -}}\n{{%- else %}}\n{{- hy_start_token ~ 'assistant' ~
hy_middle_token ~ think_begin_token -}}\n{{%- endif %}}\n{{%- endif %}}\n{{%- endif %}}\n\n", "createdAt": "2026-08-27T08:52:40.000Z", "discussionsDisabled": false, "discussionsSorting": "recently-
created", "downloads": 22357, "downloadsAllTime": 36987, "id": "tencent/Hy4-
preview", "isLikedByUser": false, "availableInferenceProviders":
[{"provider": "novita", "modelStatus": "live", "providerStatus": "live", "providerId": "tencent/hy4-
preview", "task": "conversational", "adapterWeightsPath": "model-00001-of-00131.safetensors", "features":
{"structuredOutput": true, "toolCalling": true}, "isCheapestPricingOutput": true, "isFastestThroughput": false, "isModelAuthc

```

```

{"provider":"deepinfra","modelStatus":"live","providerStatus":"live","providerId":"tencent/Hy4-
preview","task":"conversational","adapterWeightsPath":"model-00001-of-00131.safetensors","features":
{"structuredOutput":true,"toolCalling":true},"isCheapestPricingOutput":false,"isFastestThroughput":true,"isModelAuthc
generation","library_name":"transformers","librariesOther":[],"trackDownloads":true,"model-
index":null,"evalResults":[{"dataset":{"id":"Idavidrein/
gpqa"},"isBenchmark":true,"task_id":"diamond"},"value":92.3,"source":{"url":"https://huggingface.co/tencent/Hy4-
preview","name":"Model Card","isExternal":false},"filename":"eval_results/
gpqa.yaml","verified":false,"pullRequest":3,"rank":6,"label":"Diamond"},{"dataset":{"id":"ScaleAI/SWE-
bench_Pro"},"isBenchmark":true,"task_id":"SWE_Bench_Pro"},"value":65.7,"source":{"url":"https://huggingface.co/
tencent/Hy4-preview","name":"Model Card","isExternal":false},"filename":"eval_results/swe-
bench_pro.yaml","verified":false,"pullRequest":3,"rank":4,"label":"SWE Bench Pro"},{"dataset":{"id":"mercor/apex-
agents"},"isBenchmark":true,"task_id":"apex-agents"},"value":37.1,"source":{"url":"https://huggingface.co/tencent/
Hy4-preview","name":"Model Card","isExternal":false},"filename":"eval_results/apex-
agents.yaml","verified":false,"pullRequest":12,"rank":2,"label":"Apex Agents"},{"dataset":{"id":"datacurve/deep-
swe"},"isBenchmark":true,"task_id":"deep_swe"},"value":64.3,"source":{"url":"https://huggingface.co/tencent/Hy4-
preview","name":"Model Card","isExternal":false},"filename":"eval_results/deep-
swe.yaml","verified":false,"pullRequest":3,"rank":8,"label":"Deep Swe"},{"dataset":{"id":"harborframework/terminal-
bench-2.1"},"isBenchmark":true,"task_id":"terminalbench_2_1"},"value":85.4,"source":{"url":"https://huggingface.co/
tencent/Hy4-preview","name":"Model Card","isExternal":false},"filename":"eval_results/terminal-
bench-2.1.yaml","verified":false,"pullRequest":12,"rank":8,"label":"Terminalbench 2 1"},{"dataset":
{"id":"IntelligenceLab/Long-Horizon-Terminal-
Bench"},"isBenchmark":true,"task_id":"lhtb2_solved"},"value":15,"source":{"url":"https://github.com/zli12321/LHTB/
tree/lhtb-2.0#lhtb-20-results-october-2026","name":"LHTB 2.0 results","isExternal":true},"filename":"eval_results/
lhtb2_solved.yaml","verified":false,"pullRequest":14,"rank":2,"label":"Lhtb2 Solved","notes":"15/77 tasks solved at
reward >= 0.95 (pass@0.95); mean reward x100 = 53.2; LHTB 2.0 (77 tasks), terminus-2, 3-hour budget per task,
official LHTB Harbor harness"},{"dataset":{"id":"hkust-nlp/
Toolathlon"},"isBenchmark":false,"task_id":"toolathlon_verified"},"value":74.1,"source":{"url":"https://huggingface.co/
tencent/Hy4-preview","name":"tencent/Hy4-preview model card","isExternal":false},"filename":"eval_results/
tencent__Hy4-preview.yaml","verified":false,"pullRequest":13,"rank":5,"label":"Toolathlon Verified"},{"dataset":
{"id":"cais/hle"},"isBenchmark":false,"task_id":"hle"},"value":43.4,"source":{"url":"https://huggingface.co/tencent/
Hy4-preview","name":"Model Card","isExternal":false},"filename":"eval_results/
hle.yaml","verified":false,"pullRequest":12,"label":"Hle","notes":"Text-only, no tools."},"dataset":{"id":"cais/
hle"},"isBenchmark":false,"task_id":"hle"},"value":55.4,"source":{"url":"https://huggingface.co/tencent/Hy4-
preview","name":"Model Card","isExternal":false},"filename":"eval_results/
hle.yaml","verified":false,"pullRequest":12,"rank":6,"label":"Hle","notes":"Text-only, with tools."},"dataset":
{"id":"benchflow/skillsbench"},"isBenchmark":true,"task_id":"skillsbench_v1_1"},"value":62.9,"source":{"url":"https://
huggingface.co/tencent/Hy4-preview","name":"Model Card","isExternal":false},"filename":"eval_results/
skillsbench.yaml","verified":false,"pullRequest":3,"rank":2,"label":"Skillsbench V1 1"},{"dataset":{"id":"SWE-bench/
SWE-
bench_Multilingual"},"isBenchmark":true,"task_id":"swe_bench_multilingual_%_resolved"},"value":82.9,"source":
{"url":"https://huggingface.co/tencent/Hy4-preview","name":"Model
Card","isExternal":false},"filename":"eval_results/swe-
bench_multilingual.yaml","verified":false,"pullRequest":3,"rank":2,"label":"Swe Bench Multilingual
Resolved"}],"private":false,"repoType":"model","gated":false,"tags":["transformers","safetensors","hy_v4","text-
generation","hunyuan","hy4","moe","conversational","arxiv:2512.02556","arxiv:2603.12201","license:apache-2.0","eva
results","endpoints_compatible","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":"hy_v4","label":"hy_v4","type":"other","clickable":true},
{"id":"hunyuan","label":"hunyuan","type":"other","clickable":true},
{"id":"hy4","label":"hy4","type":"other","clickable":true},{"id":"moe","label":"Mixture of
Experts","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":"arxiv:2512.02556","label":"arxiv:2512.02556","type":"arxiv","extra":
{"paperTitle":"DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models"}},
{"id":"arxiv:2603.12201","label":"arxiv:2603.12201","type":"arxiv","extra":{"paperTitle":"IndexCache: Accelerating

```

Sparse Attention via Cross-Layer Index Reuse"}, {"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": {"BF16": 779930197312, "F32": 30795421}, "total": 779960992733, "sharded": true, "totalFileSize": 1559983985227}, "hasE": {"billableEntities": [], "entityName2Providers": {}, "defaultProviders": [{"isOriginalProvider": false, "name": "novita", "enabled": true, "position": 1, "isReleased": true, "accuratePricing": true}, {"isOriginalProvider": false, "name": "deepinfra", "enabled": true, "position": 15, "isReleased": true, "accuratePricing": true}]}, "Safetensors Model size 780B params

Tensor type BF16 · F32 ·

Chat template Files info

```
"eos_token": "< | hy_end:opensource | >", "pad_token": "< | hy_pad:opensource | >", "chat_template_jinja": "{#-
----- special token variables ----- #}\n{% set HYTK = 'opensource' %}\n{% set hy_start_token = '< |
hy_start{} | >'.format(HYTK) %}\n{% set hy_middle_token = '< | hy_middle{} | >'.format(HYTK) %}\n{% set
hy_end_token = '< | hy_end{} | >'.format(HYTK) %}\n{% set think_begin_token = '<think{}>'.format(HYTK)
%}\n{% set think_end_token = '</think{}>'.format(HYTK) %}\n{% set toolcalls_begin_token = '<tool_calls{}
>'.format(HYTK) %}\n{% set toolcalls_end_token = '</tool_calls{}>'.format(HYTK) %}\n{% set
toolcall_begin_token = '<tool_call{}>'.format(HYTK) %}\n{% set toolcall_end_token = '</tool_call{}>'.format(HYTK)
%}\n{% set argkey_begin_token = '<arg_key{}>'.format(HYTK) %}\n{% set argkey_end_token = '</arg_key{}
>'.format(HYTK) %}\n{% set argvalue_begin_token = '<arg_value{}>'.format(HYTK) %}\n{% set
argvalue_end_token = '</arg_value{}>'.format(HYTK) %}\n{% set toolresponse_begin_token = '<tool_response{}
>'.format(HYTK) %}\n{% set toolresponse_end_token = '</tool_response{}>'.format(HYTK) %}\n{% set
reasoning_mode_token = '< | reasoning_mode{} | >'.format(HYTK) %}\n\n#{#- ----- hyperparameters
variables ----- #}\n{% if not add_generation_prompt is defined %}\n {% set add_generation_prompt = false
%}\n{% endif %}\n{% if not preserved_thinking is defined %}\n {% if not tools %}\n {% set preserved_thinking =
false %}\n {% else %}\n {% set preserved_thinking = true %}\n {% endif %}\n{% endif %}\n{% if not
reasoning_effort is defined %}\n {% set reasoning_effort = 'high' %}\n{% elif reasoning_effort not in ['high',
'no_think'] %}\n {% if reasoning_effort is none %}\n {{ raise_exception('reasoning_effort error : None, should be
no_think/high') }}\n {% else %}\n {{ raise_exception('reasoning_effort error : ' + reasoning_effort + ', should be
no_think/high') }}\n {% endif %}\n{% endif %}\n\n{% if fallback_strategy is defined and fallback_strategy ==
'reasoning_toolcall_retry' %}\n {% set reasoning_effort = 'high' %}\n {% set add_generation_prompt = false
%}\n{% endif %}\n{% if not raw_last_assistant is defined %}\n {% set raw_last_assistant = false %}\n{% endif
%}\n\n{% macro tool_to_json(tool) -%}\n {% set ns_tool = namespace(first=true) -%}\n {{- ' ' -}}\n {% for k, v in
tool.items() -%}\n {% if k != 'defer_loading' and k != 'strict' -%}\n {% if not ns_tool.first -%}\n {{- ' ' -}}\n {% endif -%}\n
{% set ns_tool.first = false -%}\n {{- '\n' ~ k ~ '\n': ' ~ (v | tojson(ensure_ascii=False)) -}}\n {% endif -%}\n {% endfor
-%}\n {{- ' ' -}}\n{% endmacro -%}\n\n{% macro render_content(content) -%}\n {% if content is string -%}\n {{-
content -}}\n {% elif content is iterable and content is not mapping -%}\n {% for item in content -%}\n {% if item is
mapping and item.type == 'text' -%}\n {{- item.text -}}\n {% elif item is string -%}\n {{- item -}}\n {% else -%}\n {{-
item | string -}}\n {% endif -%}\n {% endfor -%}\n {% elif content is none -%}\n {{- ' ' -}}\n {% else -%}\n {{- content
| string -}}\n {% endif -%}\n{% endmacro -%}\n\n{% macro render_tools_prompt() -%}\n {{- '# Tools\n\nYou may
call one or more functions to assist with the user query.' -}}\n {{- '\n\nYou are provided with function signatures
within <tools></tools> XML tags.' -}}\n {{- '\n<tools>\n\n' -}}\n {% set tool_ns = namespace(first=true) -%}\n {% for
tool in tools -%}\n {% set t = tool['function'] if tool is mapping and 'function' in tool else tool -%}\n {% if
t.defer_loading is not defined or not t.defer_loading -%}\n {% if not tool_ns.first -%}\n {{- '\n' -}}\n {% endif -%}\n {%
set tool_ns.first = false -%}\n {{- tool_to_json(t) -}}\n {% endif -%}\n {% endfor -%}\n {{- '\n</tools>\n\n' -}}\n {{-
'For function call returns, you should first print ' ~ toolcalls_begin_token -}}\n {{- '\n\nFor each function call, you
should return object like:\n\n' -}}\n {{- toolcall_begin_token ~ '{function-name}' -}}\n {{- argkey_begin_token ~ '{arg-
key-1}' ~ argkey_end_token -}}\n {{- argvalue_begin_token ~ '{arg-value-1}' ~ argvalue_end_token -}}\n {{-
argkey_begin_token ~ '{arg-key-2}' ~ argkey_end_token -}}\n {{- argvalue_begin_token ~ '{arg-value-2}' ~
argvalue_end_token -}}\n {{- '...' -}}\n {{- toolcall_end_token -}}\n {{- '\n\nAt the end of function call returns, you
should print ' ~ toolcalls_end_token -}}\n{% endmacro -%}\n\n{% macro render_tool_response(message) -%}\n
{% set content = message['content'] -%}\n {% if content is string -%}\n {{- toolresponse_begin_token ~ content ~
toolresponse_end_token -}}\n {% elif content is iterable and content is not mapping and content and content[0] is
```

```

mapping and content[0].type == 'tool_reference' -%}\n {{- toolresponse_begin_token -}}\n {{- '<tools>\n' -}}\n {%-
for tr in content -%}\n {%- for tool in tools -%}\n {%- set t = tool['function'] if tool is mapping and 'function' in tool
else tool -%}\n {%- if t.name == tr.name -%}\n {{- tool_to_json(t) ~ '\n' -}}\n {%- endif -%}\n {%- endfor -%}\n {%-
endfor -%}\n {{- '</tools>' -}}\n {{- toolresponse_end_token -}}\n {%- elif content is iterable and content is not
mapping and content and content[0] is mapping and content[0].output is defined -%}\n {%- for tr in content -%}\n
{{- toolresponse_begin_token ~ tr.output ~ toolresponse_end_token -}}\n {%- endfor -%}\n {%- else -%}\n {{-
toolresponse_begin_token ~ render_content(content) ~ toolresponse_end_token -}}\n {%- endif -%}\n {%-
endmacro -%}\n\n{{#- consecutive tool messages are clustered into a single tool message whose content is the list
of the original tool messages; the result is written to merged_ns.messages (a macro cannot return a list) -#}\n{%-
set merged_ns = namespace(messages=[]) %}\n{%- macro merge_tool_responses(messages) -%}\n {%- set
merge_ns = namespace(result=[], cluster=[]) -%}\n {%- for message in messages -%}\n {%- if message['role'] ==
'tool' -%}\n {%- set merge_ns.cluster = merge_ns.cluster + [message] -%}\n {%- else -%}\n {%- if merge_ns.cluster
-%}\n {%- set merge_ns.result = merge_ns.result + [{'role': 'tool', 'content': merge_ns.cluster}] -%}\n {%- set
merge_ns.cluster = [] -%}\n {%- endif -%}\n {%- set merge_ns.result = merge_ns.result + [message] -%}\n {%-
endif -%}\n {%- endfor -%}\n {%- if merge_ns.cluster -%}\n {%- set merge_ns.result = merge_ns.result + [{'role':
'tool', 'content': merge_ns.cluster}] -%}\n {%- endif -%}\n {%- set merged_ns.messages = merge_ns.result -
%}\n{%- endmacro -%}\n\n{{#- render one tool cluster, reordered to follow the tool_call order of the preceding
assistant turn; any missing / ambiguous id falls back to the original tool message order for the whole cluster -
#}\n{%- macro render_tool_responses(tool_messages, tool_call_ids) -%}\n {%- set order_ns =
namespace(ordered=[], matched=true) -%}\n {%- if tool_call_ids and tool_call_ids | length == tool_messages |
length -%}\n {%- for tool_call_id in tool_call_ids -%}\n {%- set hit_ns = namespace(count=0, message=None) -%}\n
{%- for tool_message in tool_messages -%}\n {%- set message_id = tool_message['tool_call_id'] if
tool_message['tool_call_id'] is defined else tool_message['id'] -%}\n {%- if message_id is defined and message_id
== tool_call_id -%}\n {%- set hit_ns.count = hit_ns.count + 1 -%}\n {%- set hit_ns.message = tool_message -%}\n
{%- endif -%}\n {%- endfor -%}\n {%- if hit_ns.count == 1 -%}\n {%- set order_ns.ordered = order_ns.ordered +
[hit_ns.message] -%}\n {%- else -%}\n {%- set order_ns.matched = false -%}\n {%- endif -%}\n {%- endfor -%}\n
{%- else -%}\n {%- set order_ns.matched = false -%}\n {%- endif -%}\n {%- for tool_message in (order_ns.ordered
if order_ns.matched else tool_messages) -%}\n {{- render_tool_response(tool_message) -}}\n {%- endfor -%}\n{%-
endmacro -%}\n\n{%- set _ = merge_tool_responses(messages) %}\n{%- set normed_messages =
merged_ns.messages %}\n\n{%- set ns = namespace(last_user_index=-1, has_leading_system=false) %}\n{%- if
normed_messages and normed_messages[0].role == 'system' %}\n {%- set ns.has_leading_system = true
%}\n{%- endif %}\n{%- for message in normed_messages %}\n {%- if message['role'] == 'user' %}\n {%- set
ns.last_user_index = loop.index0 %}\n {%- endif %}\n{%- endfor %}\n\n{{#- no leading system: synthesize one for
tools and/or reasoning_mode -#}\n{%- if not ns.has_leading_system %}\n {{- hy_start_token ~ 'system' ~
hy_middle_token -}}\n {%- if tools %}\n {{- render_tools_prompt() -}}\n {%- endif %}\n {{- reasoning_mode_token ~
'reasoning_effort:' ~ reasoning_effort -}}\n {{- hy_end_token -}}\n{%- endif %}\n\n{%- set last_ns =
namespace(last_is_assistant=false) %}\n{%- set prev_ns = namespace(tool_call_ids=[]) %}\n{%- for message in
normed_messages %}\n {%- if message['role'] == 'system' %}\n {{- hy_start_token ~ 'system' ~ hy_middle_token
-}}\n {{#- tools / reasoning_mode only attach to the leading system (messages[0]) -#}\n {%- if loop.first %}\n {%- if
tools %}\n {{- render_tools_prompt() -}}\n {%- endif %}\n {%- set content = render_content(message['content']) -
%}\n {%- if tools and content -%}\n {{- '\n\n' -}}\n {%- endif -%}\n {{- content -}}\n {{- reasoning_mode_token ~
'reasoning_effort:' ~ reasoning_effort -}}\n {%- else %}\n {{- render_content(message['content']) -}}\n {%- endif
%}\n {{- hy_end_token -}}\n {%- elif message['role'] == 'user' %}\n {{- hy_start_token ~ 'user' ~ hy_middle_token
-}}\n {{- render_content(message['content']) -}}\n {{- hy_end_token -}}\n {%- elif message['role'] == 'assistant' %}\n
{%- set content_body = render_content(message['content']) -%}\n {{#- 'reasoning' takes precedence, fall back to
'reasoning_content' when it is empty -#}\n {%- set reasoning_text = "" %}\n {%- if message['reasoning'] is defined
and message['reasoning'] is string and message['reasoning'] %}\n {%- set reasoning_text = message['reasoning']
%}\n {%- elif message['reasoning_content'] is defined and message['reasoning_content'] is string and
message['reasoning_content'] %}\n {%- set reasoning_text = message['reasoning_content'] %}\n {%- endif %}\n
{{#- no_think overrides preserved_thinking: history is always emitted with empty think tags -#}\n {%- if
reasoning_effort != 'no_think' and (preserved_thinking or loop.index0 > ns.last_user_index) and reasoning_text
%}\n {%- set content = think_begin_token ~ reasoning_text ~ think_end_token ~ content_body %}\n {%- else %}\n
{%- set content = think_begin_token ~ think_end_token ~ content_body %}\n {%- endif %}\n {{- hy_start_token ~
'assistant' ~ hy_middle_token -}}\n {%- if message['tool_calls'] is defined and message['tool_calls'] %}\n {{#-
remember the tool_call ids of this turn to order the following tool responses; an incomplete id set means 'do not
reorder' -#}\n {%- set ids_ns = namespace(ids=[], complete=true) %}\n {%- for tool in message['tool_calls'] %}\n

```

```

{% set tool_call_id = tool['id'] if tool['id'] is defined and tool['id'] else tool['tool_call_id'] %}
{% if tool_call_id is defined and tool_call_id %}
{% set ids_ns.ids = ids_ns.ids + [tool_call_id] %}
{% else %}
{% set ids_ns.complete = false %}
{% endif %}
{% endfor %}
{% set prev_ns.tool_call_ids = ids_ns.ids if ids_ns.complete else [] %}
{{ content }}
{{ toolcalls_begin_token }}
{% for tool in message['tool_calls'] - %}
{% set func = tool['function'] if tool is mapping and 'function' in tool else tool - %}
{% set arguments = func['arguments'] - %}
{{ toolcall_begin_token ~ func['name'] - }}
{% for key, value in arguments.items() - %}
{{ argkey_begin_token ~ key ~ argkey_end_token - }}
{% if value is not string - %}
{% set value = value | tojson(ensure_ascii=False) - %}
{% endif - %}
{{ argvalue_begin_token ~ value ~ argvalue_end_token - }}
{% endfor - %}
{{ toolcall_end_token }}
{% endfor - %}
{{ toolcalls_end_token }}
{% else %}
{% if loop.last and raw_last_assistant %}
{{ content_body }}
{% else %}
{{ content }}
{% endif %}
{% endif %}
{{ # continuation / prefill: last assistant is still open, do not close with hy_end - # }}
{% if not (loop.last and raw_last_assistant) %}
{{ hy_end_token }}
{% endif %}
{% elif message['role'] == 'tool' %}
{{ hy_start_token ~ 'tool' ~ hy_middle_token - }}
{{ render_tool_responses(message['content'], prev_ns.tool_call_ids) - }}
{{ hy_end_token - }}
{% else %}
{{ hy_start_token ~ message['role'] ~ hy_middle_token - }}
{{ render_content(message['content']) - }}
{{ hy_end_token - }}
{% endif %}
{% if message['role'] != 'assistant' or not (message['tool_calls'] is defined and message['tool_calls']) %}
{% set prev_ns.tool_call_ids = [] %}
{% endif %}
{% if loop.last and message['role'] == 'assistant' %}
{% set last_ns.last_is_assistant = true %}
{% endif %}
{% endfor %}
{% if add_generation_prompt %}
{% if not last_ns.last_is_assistant %}
{% if reasoning_effort == 'no_think' %}
{{ hy_start_token ~ 'assistant' ~ hy_middle_token ~ think_begin_token ~ think_end_token - }}
{% else %}
{{ hy_start_token ~ 'assistant' ~ hy_middle_token ~ think_begin_token - }}
{% endif %}
{% endif %}
{{ "createdAt":"2026-08-27T08:52:40.000Z","discussionsDisabled":false,"discussionsSorting":"recently-created","downloads":22357,"downloadsAllTime":36987,"id":"tencent/Hy4-preview","isLikedByUser":false,"availableInferenceProviders":
[{"provider":"novita","modelStatus":"live","providerStatus":"live","providerId":"tencent/hy4-preview","task":"conversational","adapterWeightsPath":"model-00001-of-00131.safetensors","features":{"structuredOutput":true,"toolCalling":true},"isCheapestPricingOutput":true,"isFastestThroughput":false,"isModelAuthc
{"provider":"deepinfra","modelStatus":"live","providerStatus":"live","providerId":"tencent/Hy4-preview","task":"conversational","adapterWeightsPath":"model-00001-of-00131.safetensors","features":{"structuredOutput":true,"toolCalling":true},"isCheapestPricingOutput":false,"isFastestThroughput":true,"isModelAuthc
generation","library_name":"transformers","librariesOther":[],"trackDownloads":true,"model-index":null,"evalResults":[{"dataset":{"id":"ldavidrein/gpqa","isBenchmark":true,"task_id":"diamond"},"value":92.3,"source":{"url":"https://huggingface.co/tencent/Hy4-preview","name":"Model Card","isExternal":false},"filename":"eval_results/gpqa.yaml","verified":false,"pullRequest":3,"rank":6,"label":"Diamond"},{"dataset":{"id":"ScaleAI/SWE-bench_Pro","isBenchmark":true,"task_id":"SWE_Bench_Pro"},"value":65.7,"source":{"url":"https://huggingface.co/tencent/Hy4-preview","name":"Model Card","isExternal":false},"filename":"eval_results/swe-bench_pro.yaml","verified":false,"pullRequest":3,"rank":4,"label":"SWE Bench Pro"},{"dataset":{"id":"mercor/apex-agents","isBenchmark":true,"task_id":"apex-agents"},"value":37.1,"source":{"url":"https://huggingface.co/tencent/Hy4-preview","name":"Model Card","isExternal":false},"filename":"eval_results/apex-agents.yaml","verified":false,"pullRequest":12,"rank":2,"label":"Apex Agents"},{"dataset":{"id":"datacurve/deep-swe","isBenchmark":true,"task_id":"deep_swe"},"value":64.3,"source":{"url":"https://huggingface.co/tencent/Hy4-preview","name":"Model Card","isExternal":false},"filename":"eval_results/deep-swe.yaml","verified":false,"pullRequest":3,"rank":8,"label":"Deep Swe"},{"dataset":{"id":"harborframework/terminal-bench-2.1","isBenchmark":true,"task_id":"terminalbench_2_1"},"value":85.4,"source":{"url":"https://huggingface.co/tencent/Hy4-preview","name":"Model Card","isExternal":false},"filename":"eval_results/terminal-bench-2.1.yaml","verified":false,"pullRequest":12,"rank":8,"label":"Terminalbench 2 1"},{"dataset":{"id":"IntelligenceLab/Long-Horizon-Terminal-Bench","isBenchmark":true,"task_id":"lhtb2_solved"},"value":15,"source":{"url":"https://github.com/zli12321/LHTB/tree/lhtb-2.0#lhtb-20-results-october-2026","name":"LHTB 2.0 results","isExternal":true},"filename":"eval_results/lhtb2_solved.yaml","verified":false,"pullRequest":14,"rank":2,"label":"Lhtb2 Solved","notes":"15/77 tasks solved at reward >= 0.95 (pass@0.95); mean reward x100 = 53.2; LHTB 2.0 (77 tasks), terminus-2, 3-hour budget per task, official LHTB Harbor harness"},{"dataset":{"id":"hkust-nlp/Toolathlon","isBenchmark":false,"task_id":"toolathlon_verified"},"value":74.1,"source":{"url":"https://huggingface.co/tencent/Hy4-preview","name":"tencent/Hy4-preview model card","isExternal":false},"filename":"eval_results/tencent__Hy4-preview.yaml","verified":false,"pullRequest":13,"rank":5,"label":"Toolathlon Verified"},{"dataset":

```

```
{
  "id": "cais/hle",
  "isBenchmark": false,
  "task_id": "hle",
  "value": 43.4,
  "source": {
    "url": "https://huggingface.co/tencent/Hy4-preview",
    "name": "Model Card",
    "isExternal": false,
    "filename": ".eval_results/hle.yaml",
    "verified": false,
    "pullRequest": 12,
    "label": "Hle",
    "notes": "Text-only, no tools."
  },
  {"dataset": {"id": "cais/hle", "isBenchmark": false, "task_id": "hle", "value": 55.4, "source": {"url": "https://huggingface.co/tencent/Hy4-preview", "name": "Model Card", "isExternal": false, "filename": ".eval_results/hle.yaml", "verified": false, "pullRequest": 12, "rank": 6, "label": "Hle", "notes": "Text-only, with tools."}}, {"dataset": {"id": "benchflow/skillsbench", "isBenchmark": true, "task_id": "skillsbench_v1_1", "value": 62.9, "source": {"url": "https://huggingface.co/tencent/Hy4-preview", "name": "Model Card", "isExternal": false, "filename": ".eval_results/skillsbench.yaml", "verified": false, "pullRequest": 3, "rank": 2, "label": "Skillsbench V1 1"}, {"dataset": {"id": "SWE-bench/SWE-bench_Multilingual", "isBenchmark": true, "task_id": "swe_bench_multilingual_%_resolved", "value": 82.9, "source": {"url": "https://huggingface.co/tencent/Hy4-preview", "name": "Model Card", "isExternal": false, "filename": ".eval_results/swe-bench_multilingual.yaml", "verified": false, "pullRequest": 3, "rank": 2, "label": "Swe Bench Multilingual Resolved"}}, {"private": false, "repoType": "model", "gated": false, "tags": ["transformers", "safetensors", "hy_v4", "text-generation", "hunyuan", "hy4", "moe", "conversational", "arxiv:2512.02556", "arxiv:2603.12201", "license:apache-2.0", "eval_results", "endpoints_compatible", "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": "hy_v4", "label": "hy_v4", "type": "other", "clickable": true}, {"id": "hunyuan", "label": "hunyuan", "type": "other", "clickable": true}, {"id": "hy4", "label": "hy4", "type": "other", "clickable": true}, {"id": "moe", "label": "Mixture of Experts", "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": "arxiv:2512.02556", "label": "arxiv:2512.02556", "type": "arxiv", "extra": {"paperTitle": "DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models"}}, {"id": "arxiv:2603.12201", "label": "arxiv:2603.12201", "type": "arxiv", "extra": {"paperTitle": "IndexCache: Accelerating Sparse Attention via Cross-Layer Index Reuse"}}, {"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": {"BF16": 779930197312, "F32": 30795421}, "total": 779960992733, "sharded": true, "totalFileSize": 1559983985227}, "hasE": {}, "inferenceContextData": {"billableEntities": [], "entityName2Providers": {}, "defaultProviders": [{"isOriginalProvider": false, "name": "novita", "enabled": true, "position": 1, "isReleased": true, "accuratePricing": true}, {"isOriginalProvider": false, "name": "deepinfra", "enabled": true, "position": 15, "isReleased": true, "accuratePricing": true}], "url": "/spaces/huggingface/InferenceSupport/discussions/12026", "numReactions": 18, "isLoggedIn": false, "canWrite": false, "examples": [{"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."}], "initialSelectedProvider": "novita", "widgetType": "conversational", "isMaximized": false, "isMac": true}]>
```

Text Generation

Examples

Input a message to start chatting with tencent/Hy4-preview .

Send

View Code Snippets

Compare providers

Model tree for tencent/Hy4-preview

Quantizations

7 models

Spaces using tencent/Hy4-preview 7

Collection including tencent/Hy4-preview 2 items • Updated about 6 hours ago • 15

Papers for tencent/Hy4-preview Paper • 2603.12201 • Published Mar 12 • 69 Paper • 2512.02556 • Published Dec 2, 2025 • 274

```

eos_token": "< | hy_end:">" , "pad_token": "< | hy_pad:">" , "chat_template_jinja": "#-
----- special token variables ----- #\n{% set HYTK = 'openseource' %}\n{% set hy_start_token = '< |
hy_start{'>' .format(HYTK) %}\n{% set hy_middle_token = '< | hy_middle{'>' .format(HYTK) %}\n{% set
hy_end_token = '< | hy_end{'>' .format(HYTK) %}\n{% set think_begin_token = '<think{'>' .format(HYTK)
%}\n{% set think_end_token = '</think{'>' .format(HYTK) %}\n{% set toolcalls_begin_token = '<tool_calls{'
>' .format(HYTK) %}\n{% set toolcalls_end_token = '</tool_calls{'>' .format(HYTK) %}\n{% set
toolcall_begin_token = '<tool_call{'>' .format(HYTK) %}\n{% set toolcall_end_token = '</tool_call{'>' .format(HYTK)
%}\n{% set argkey_begin_token = '<arg_key{'>' .format(HYTK) %}\n{% set argkey_end_token = '</arg_key{'
>' .format(HYTK) %}\n{% set argvalue_begin_token = '<arg_value{'>' .format(HYTK) %}\n{% set
argvalue_end_token = '</arg_value{'>' .format(HYTK) %}\n{% set toolresponse_begin_token = '<tool_response{'
>' .format(HYTK) %}\n{% set toolresponse_end_token = '</tool_response{'>' .format(HYTK) %}\n{% set
reasoning_mode_token = '< | reasoning_mode{'>' .format(HYTK) %}\n\n#- ----- hyperparameters
variables ----- #\n{% if not add_generation_prompt is defined %}\n {% set add_generation_prompt = false
%}\n{% endif %}\n{% if not preserved_thinking is defined %}\n {% if not tools %}\n {% set preserved_thinking =
false %}\n {% else %}\n {% set preserved_thinking = true %}\n {% endif %}\n{% endif %}\n\n{% if not
reasoning_effort is defined %}\n {% set reasoning_effort = 'high' %}\n{% elif reasoning_effort not in ['high',
'no_think'] %}\n {% if reasoning_effort is none %}\n {{ raise_exception('reasoning_effort error : None, should be
no_think/high') }}\n {% else %}\n {{ raise_exception('reasoning_effort error : ' + reasoning_effort + ', should be
no_think/high') }}\n {% endif %}\n{% endif %}\n\n{% if fallback_strategy is defined and fallback_strategy ==
'reasoning_toolcall_retry' %}\n {% set reasoning_effort = 'high' %}\n {% set add_generation_prompt = false
%}\n{% endif %}\n{% if not raw_last_assistant is defined %}\n {% set raw_last_assistant = false %}\n{% endif
%}\n\n{% macro tool_to_json(tool) -%}\n {% set ns_tool = namespace(first=true) -%}\n {{ '-' -}}\n {% for k, v in
tool.items() -%}\n {% if k != 'defer_loading' and k != 'strict' -%}\n {% if not ns_tool.first -%}\n {{ '-', -}}\n {% endif -%}\n
{% set ns_tool.first = false -%}\n {{ '-' ~ k ~ ':' ~ (v | tojson(ensure_ascii=False)) -}}\n {% endif -%}\n {% endfor
-%}\n {{ '-' -}}\n {% endmacro -%}\n\n{% macro render_content(content) -%}\n {% if content is string -%}\n {{-
content -}}\n {% elif content is iterable and content is not mapping -%}\n {% for item in content -%}\n {% if item is
mapping and item.type == 'text' -%}\n {{- item.text -}}\n {% elif item is string -%}\n {{- item -}}\n {% else -%}\n {{-
item | string -}}\n {% endif -%}\n {% endfor -%}\n {% elif content is none -%}\n {{- '' -}}\n {% else -%}\n {{- content
| string -}}\n {% endif -%}\n {% endmacro -%}\n\n{% macro render_tools_prompt() -%}\n {{- '# Tools\n\nYou may
call one or more functions to assist with the user query.' -}}\n {{- '\n\nYou are provided with function signatures
within <tools></tools> XML tags:' -}}\n {{- '\n<tools>\n' -}}\n {% set tool_ns = namespace(first=true) -%}\n {% for
tool in tools -%}\n {% set t = tool['function'] if tool is mapping and 'function' in tool else tool -%}\n {% if
t.defer_loading is not defined or not t.defer_loading -%}\n {% if not tool_ns.first -%}\n {{- '\n' -}}\n {% endif -%}\n {%
set tool_ns.first = false -%}\n {{- tool_to_json(t) -}}\n {% endif -%}\n {% endfor -%}\n {{- '\n</tools>\n\n' -}}\n {{-
'For function call returns, you should first print ' ~ toolcalls_begin_token -}}\n {{- '\nFor each function call, you
should return object like:\n' -}}\n {{- toolcall_begin_token ~ '{function-name}' -}}\n {{- argkey_begin_token ~ '{arg-
key-1}' ~ argkey_end_token -}}\n {{- argvalue_begin_token ~ '{arg-value-1}' ~ argvalue_end_token -}}\n {{-
argkey_begin_token ~ '{arg-key-2}' ~ argkey_end_token -}}\n {{- argvalue_begin_token ~ '{arg-value-2}' ~
argvalue_end_token -}}\n {{- '...' -}}\n {{- toolcall_end_token -}}\n {{- '\nAt the end of function call returns, you
should print ' ~ toolcalls_end_token -}}\n{% endmacro -%}\n\n{% macro render_tool_response(message) -%}\n
{% set content = message['content'] -%}\n {% if content is string -%}\n {{- toolresponse_begin_token ~ content ~
toolresponse_end_token -}}\n {% elif content is iterable and content is not mapping and content and content[0] is
mapping and content[0].type == 'tool_reference' -%}\n {{- toolresponse_begin_token -}}\n {{- '<tools>\n' -}}\n {%
for tr in content -%}\n {% for tool in tools -%}\n {% set t = tool['function'] if tool is mapping and 'function' in tool
else tool -%}\n {% if t.name == tr.name -%}\n {{- tool_to_json(t) ~ '\n' -}}\n {% endif -%}\n {% endfor -%}\n {%
endfor -%}\n {{- '</tools>' -}}\n {{- toolresponse_end_token -}}\n {% elif content is iterable and content is not
mapping and content and content[0] is mapping and content[0].output is defined -%}\n {% for tr in content -%}\n
{{- toolresponse_begin_token ~ tr.output ~ toolresponse_end_token -}}\n {% endfor -%}\n {% else -%}\n {{-
toolresponse_begin_token ~ render_content(content) ~ toolresponse_end_token -}}\n {% endif -%}\n{%

```

```

endmacro -%}\n\n{%- consecutive tool messages are clustered into a single tool message whose content is the list
of the original tool messages; the result is written to merged_ns.messages (a macro cannot return a list) -#\n{%-
set merged_ns = namespace(messages=[]) %}\n{%- macro merge_tool_responses(messages) -%}\n{%- set
merge_ns = namespace(result=[], cluster=[]) -%}\n{%- for message in messages -%}\n{%- if message['role'] ==
'tool' -%}\n{%- set merge_ns.cluster = merge_ns.cluster + [message] -%}\n{%- else -%}\n{%- if merge_ns.cluster
-%}\n{%- set merge_ns.result = merge_ns.result + [{'role': 'tool', 'content': merge_ns.cluster}] -%}\n{%- set
merge_ns.cluster = [] -%}\n{%- endif -%}\n{%- set merge_ns.result = merge_ns.result + [message] -%}\n{%-
endif -%}\n{%- endfor -%}\n{%- if merge_ns.cluster -%}\n{%- set merge_ns.result = merge_ns.result + [{'role':
'tool', 'content': merge_ns.cluster}] -%}\n{%- endif -%}\n{%- set merged_ns.messages = merge_ns.result -
%}\n{%- endmacro -%}\n\n{%- render one tool cluster, reordered to follow the tool_call order of the preceding
assistant turn; any missing / ambiguous id falls back to the original tool message order for the whole cluster -
#\n{%- macro render_tool_responses(tool_messages, tool_call_ids) -%}\n{%- set order_ns =
namespace(ordered=[], matched=true) -%}\n{%- if tool_call_ids and tool_call_ids | length == tool_messages |
length -%}\n{%- for tool_call_id in tool_call_ids -%}\n{%- set hit_ns = namespace(count=0, message=None) -%}\n
{%- for tool_message in tool_messages -%}\n{%- set message_id = tool_message['tool_call_id'] if
tool_message['tool_call_id'] is defined else tool_message['id'] -%}\n{%- if message_id is defined and message_id
== tool_call_id -%}\n{%- set hit_ns.count = hit_ns.count + 1 -%}\n{%- set hit_ns.message = tool_message -%}\n
{%- endif -%}\n{%- endfor -%}\n{%- if hit_ns.count == 1 -%}\n{%- set order_ns.ordered = order_ns.ordered +
[hit_ns.message] -%}\n{%- else -%}\n{%- set order_ns.matched = false -%}\n{%- endif -%}\n{%- endfor -%}\n
{%- else -%}\n{%- set order_ns.matched = false -%}\n{%- endif -%}\n{%- for tool_message in (order_ns.ordered
if order_ns.matched else tool_messages) -%}\n{%- render_tool_response(tool_message) -%}\n{%- endfor -%}\n{%-
endmacro -%}\n\n{%- set _ = merge_tool_responses(messages) %}\n{%- set normed_messages =
merged_ns.messages %}\n\n{%- set ns = namespace(last_user_index=-1, has_leading_system=false) %}\n{%- if
normed_messages and normed_messages[0].role == 'system' %}\n{%- set ns.has_leading_system = true
%}\n{%- endif %}\n{%- for message in normed_messages %}\n{%- if message['role'] == 'user' %}\n{%- set
ns.last_user_index = loop.index0 %}\n{%- endif %}\n{%- endfor %}\n\n{%- no leading system: synthesize one for
tools and/or reasoning_mode -#\n{%- if not ns.has_leading_system %}\n{%- hy_start_token ~ 'system' ~
hy_middle_token -%}\n{%- if tools %}\n{%- render_tools_prompt() -%}\n{%- endif %}\n{%- reasoning_mode_token ~
'reasoning_effort:' ~ reasoning_effort -%}\n{%- hy_end_token -%}\n{%- endif %}\n\n{%- set last_ns =
namespace(last_is_assistant=false) %}\n{%- set prev_ns = namespace(tool_call_ids=[]) %}\n{%- for message in
normed_messages %}\n{%- if message['role'] == 'system' %}\n{%- hy_start_token ~ 'system' ~ hy_middle_token
-%}\n{%- # tools / reasoning_mode only attach to the leading system (messages[0]) -#\n{%- if loop.first %}\n{%- if
tools %}\n{%- render_tools_prompt() -%}\n{%- endif %}\n{%- set content = render_content(message['content']) -
%}\n{%- if tools and content -%}\n{%- '\n\n' -%}\n{%- endif -%}\n{%- content -%}\n{%- reasoning_mode_token ~
'reasoning_effort:' ~ reasoning_effort -%}\n{%- else %}\n{%- render_content(message['content']) -%}\n{%- endif
%}\n{%- hy_end_token -%}\n{%- elif message['role'] == 'user' %}\n{%- hy_start_token ~ 'user' ~ hy_middle_token
-%}\n{%- render_content(message['content']) -%}\n{%- hy_end_token -%}\n{%- elif message['role'] == 'assistant' %}\n
{%- set content_body = render_content(message['content']) -%}\n{%- # 'reasoning' takes precedence, fall back to
'reasoning_content' when it is empty -#\n{%- set reasoning_text = " %}\n{%- if message['reasoning'] is defined
and message['reasoning'] is string and message['reasoning'] %}\n{%- set reasoning_text = message['reasoning']
%}\n{%- elif message['reasoning_content'] is defined and message['reasoning_content'] is string and
message['reasoning_content'] %}\n{%- set reasoning_text = message['reasoning_content'] %}\n{%- endif %}\n
{%- # no_think overrides preserved_thinking: history is always emitted with empty think tags -#\n{%- if
reasoning_effort != 'no_think' and (preserved_thinking or loop.index0 > ns.last_user_index) and reasoning_text
%}\n{%- set content = think_begin_token ~ reasoning_text ~ think_end_token ~ content_body %}\n{%- else %}\n
{%- set content = think_begin_token ~ think_end_token ~ content_body %}\n{%- endif %}\n{%- hy_start_token ~
'assistant' ~ hy_middle_token -%}\n{%- if message['tool_calls'] is defined and message['tool_calls'] %}\n{%-
remember the tool_call ids of this turn to order the following tool responses; an incomplete id set means 'do not
reorder' -#\n{%- set ids_ns = namespace(ids=[], complete=true) %}\n{%- for tool in message['tool_calls'] %}\n
{%- set tool_call_id = tool['id'] if tool['id'] is defined and tool['id'] else tool['tool_call_id'] %}\n{%- if tool_call_id is
defined and tool_call_id %}\n{%- set ids_ns.ids = ids_ns.ids + [tool_call_id] %}\n{%- else %}\n{%- set
ids_ns.complete = false %}\n{%- endif %}\n{%- endfor %}\n{%- set prev_ns.tool_call_ids = ids_ns.ids if
ids_ns.complete else [] %}\n{%- content -%}\n{%- toolcalls_begin_token -%}\n{%- for tool in message['tool_calls'] -
%}\n{%- set func = tool['function'] if tool is mapping and 'function' in tool else tool -%}\n{%- set arguments =
func['arguments'] -%}\n{%- toolcall_begin_token ~ func['name'] -%}\n{%- for key, value in arguments.items() -%}\n
{%- argkey_begin_token ~ key ~ argkey_end_token -%}\n{%- if value is not string -%}\n{%- set value = value |

```

```

tojson(ensure_ascii=False) -%}\n {%\n -endif -%}\n {{\n -argvalue_begin_token ~ value ~ argvalue_end_token -}}\n
{{\n -endifor -%}\n {{\n -toolcall_end_token -}}\n {{\n -endifor -%}\n {{\n -toolcalls_end_token -}}\n {{\n -else %}\n {{\n -if
loop.last and raw_last_assistant %}\n {{\n -content_body -}}\n {{\n -else %}\n {{\n -content -}}\n {{\n -endif %}\n {{\n -endif
%}\n {{\n -# continuation / prefill: last assistant is still open, do not close with hy_end -#\n {{\n -if not (loop.last and
raw_last_assistant) %}\n {{\n -hy_end_token -}}\n {{\n -endif %}\n {{\n -elif message['role'] == 'tool' %}\n {{\n -
hy_start_token ~ 'tool' ~ hy_middle_token -}}\n {{\n -render_tool_responses(message['content'],
prev_ns.tool_call_ids) -}}\n {{\n -hy_end_token -}}\n {{\n -else %}\n {{\n -hy_start_token ~ message['role'] ~
hy_middle_token -}}\n {{\n -render_content(message['content']) -}}\n {{\n -hy_end_token -}}\n {{\n -endif %}\n {{\n -if
message['role'] != 'assistant' or not (message['tool_calls'] is defined and message['tool_calls']) %}\n {{\n -set
prev_ns.tool_call_ids = [] %}\n {{\n -endif %}\n {{\n -if loop.last and message['role'] == 'assistant' %}\n {{\n -set
last_ns.last_is_assistant = true %}\n {{\n -endif %}\n {{\n -endifor %}\n {{\n -if add_generation_prompt %}\n {{\n -if not
last_ns.last_is_assistant %}\n {{\n -if reasoning_effort == 'no_think' %}\n {{\n -hy_start_token ~ 'assistant' ~
hy_middle_token ~ think_begin_token ~ think_end_token -}}\n {{\n -else %}\n {{\n -hy_start_token ~ 'assistant' ~
hy_middle_token ~ think_begin_token -}}\n {{\n -endif %}\n {{\n -endif %}\n {{\n -endif
%}\n {{\n -created_at:"2026-08-27T08:52:40.000Z",discussions_disabled:false,discussions_sorting:"recently-
created",downloads:22357,downloads_all_time:36987,id:"tencent/Hy4-
preview",is_liked_by_user:false,available_inference_providers":
[{"provider":"novita","model_status":"live","provider_status":"live","provider_id":"tencent/hy4-
preview","task":"conversational","adapter_weights_path":"model-00001-of-00131.safetensors","features":
{"structured_output":true,"tool_calling":true,"is_cheapest_pricing_output":true,"is_fastest_throughput":false,"is_model_authc
{"provider":"deepinfra","model_status":"live","provider_status":"live","provider_id":"tencent/Hy4-
preview","task":"conversational","adapter_weights_path":"model-00001-of-00131.safetensors","features":
{"structured_output":true,"tool_calling":true,"is_cheapest_pricing_output":false,"is_fastest_throughput":true,"is_model_authc
generation","library_name":"transformers","libraries_other":[],"track_downloads":true,"model-
index":null,"eval_results":{"dataset":{"id":"ldavidrein/
gpqa","is_benchmark":true,"task_id":"diamond"},"value":92.3,"source":{"url":"https://huggingface.co/tencent/Hy4-
preview","name":"Model Card","is_external":false},"filename":"eval_results/
gpqa.yaml","verified":false,"pull_request":3,"rank":6,"label":"Diamond"},"dataset":{"id":"ScaleAI/SWE-
bench_Pro","is_benchmark":true,"task_id":"SWE_Bench_Pro"},"value":65.7,"source":{"url":"https://huggingface.co/
tencent/Hy4-preview","name":"Model Card","is_external":false},"filename":"eval_results/swe-
bench_pro.yaml","verified":false,"pull_request":3,"rank":4,"label":"SWE Bench Pro"},"dataset":{"id":"mercor/apex-
agents","is_benchmark":true,"task_id":"apex-agents"},"value":37.1,"source":{"url":"https://huggingface.co/tencent/
Hy4-preview","name":"Model Card","is_external":false},"filename":"eval_results/apex-
agents.yaml","verified":false,"pull_request":12,"rank":2,"label":"Apex Agents"},"dataset":{"id":"datacurve/deep-
swe","is_benchmark":true,"task_id":"deep_swe"},"value":64.3,"source":{"url":"https://huggingface.co/tencent/Hy4-
preview","name":"Model Card","is_external":false},"filename":"eval_results/deep-
swe.yaml","verified":false,"pull_request":3,"rank":8,"label":"Deep Swe"},"dataset":{"id":"harborframework/terminal-
bench-2.1","is_benchmark":true,"task_id":"terminalbench_2_1"},"value":85.4,"source":{"url":"https://huggingface.co/
tencent/Hy4-preview","name":"Model Card","is_external":false},"filename":"eval_results/terminal-
bench-2.1.yaml","verified":false,"pull_request":12,"rank":8,"label":"Terminalbench 2 1"},"dataset":
{"id":"IntelligenceLab/Long-Horizon-Terminal-
Bench","is_benchmark":true,"task_id":"lhtb2_solved"},"value":15,"source":{"url":"https://github.com/zli12321/LHTB/
tree/lhtb-2.0#lhtb-20-results-october-2026","name":"LHTB 2.0 results","is_external":true},"filename":"eval_results/
lhtb2_solved.yaml","verified":false,"pull_request":14,"rank":2,"label":"Lhtb2 Solved","notes":"15/77 tasks solved at
reward >= 0.95 (pass@0.95); mean reward x100 = 53.2; LHTB 2.0 (77 tasks), terminus-2, 3-hour budget per task,
official LHTB Harbor harness"},"dataset":{"id":"hkust-nlp/
Toolathlon","is_benchmark":false,"task_id":"toolathlon_verified"},"value":74.1,"source":{"url":"https://huggingface.co/
tencent/Hy4-preview","name":"tencent/Hy4-preview model card","is_external":false},"filename":"eval_results/
tencent__Hy4-preview.yaml","verified":false,"pull_request":13,"rank":5,"label":"Toolathlon Verified"},"dataset":
{"id":"cais/hle","is_benchmark":false,"task_id":"hle"},"value":43.4,"source":{"url":"https://huggingface.co/tencent/
Hy4-preview","name":"Model Card","is_external":false},"filename":"eval_results/
hle.yaml","verified":false,"pull_request":12,"label":"Hle","notes":"Text-only, no tools."},"dataset":{"id":"cais/
hle","is_benchmark":false,"task_id":"hle"},"value":55.4,"source":{"url":"https://huggingface.co/tencent/Hy4-
preview","name":"Model Card","is_external":false},"filename":"eval_results/
hle.yaml","verified":false,"pull_request":12,"rank":6,"label":"Hle","notes":"Text-only, with tools."},"dataset":
{"id":"benchflow/skillsbench","is_benchmark":true,"task_id":"skillsbench_v1_1"},"value":62.9,"source":{"url":"https://

```

huggingface.co/tencent/Hy4-preview", "name": "Model Card", "isExternal": false, "filename": ".eval_results/skillsbench.yaml", "verified": false, "pullRequest": 3, "rank": 2, "label": "Skillsbench V1 1"}, {"dataset": {"id": "SWE-bench/SWE-bench_Multilingual", "isBenchmark": true, "task_id": "swe_bench_multilingual_%_resolved"}, "value": 82.9, "source": {"url": "https://huggingface.co/tencent/Hy4-preview", "name": "Model Card", "isExternal": false, "filename": ".eval_results/swe-bench_multilingual.yaml", "verified": false, "pullRequest": 3, "rank": 2, "label": "Swe Bench Multilingual Resolved"}}, {"private": false, "repoType": "model", "gated": false, "tags": ["transformers", "safetensors", "hy_v4", "text-generation", "hunyuan", "hy4", "moe", "conversational", "arxiv:2512.02556", "arxiv:2603.12201", "license:apache-2.0", "evaluation results", "endpoints_compatible", "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": "hy_v4", "label": "hy_v4", "type": "other", "clickable": true}, {"id": "hunyuan", "label": "hunyuan", "type": "other", "clickable": true}, {"id": "hy4", "label": "hy4", "type": "other", "clickable": true}, {"id": "moe", "label": "Mixture of Experts", "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": "arxiv:2512.02556", "label": "arxiv:2512.02556", "type": "arxiv", "extra": {"paperTitle": "DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models"}}, {"id": "arxiv:2603.12201", "label": "arxiv:2603.12201", "type": "arxiv", "extra": {"paperTitle": "IndexCache: Accelerating Sparse Attention via Cross-Layer Index Reuse"}}, {"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": {"BF16": 779930197312, "F32": 30795421}, "total": 779960992733, "sharded": true, "totalFileSize": 1559983985227}, "hasEvaluation results": "Idavidrein/gpqa · Diamond View evaluation results leaderboard 92.3", "ScaleAI/SWE-bench_Pro · SWE Bench Pro View evaluation results leaderboard 65.7", "mercor/apex-agents · Apex Agents View evaluation results leaderboard 37.1", "datacurve/deep-swe · Deep Swe View evaluation results leaderboard 64.3", "harborframework/terminal-bench-2.1 · Terminalbench 2 1 View evaluation results leaderboard 85.4", "IntelligenceLab/Long-Horizon-Terminal-Bench · Lhtb2 Solved View evaluation results source leaderboard 15 *", "hkust-nlp/Toolathlon · Toolathlon Verified View evaluation results 74.1", "Expand 3 benchmarks"}]