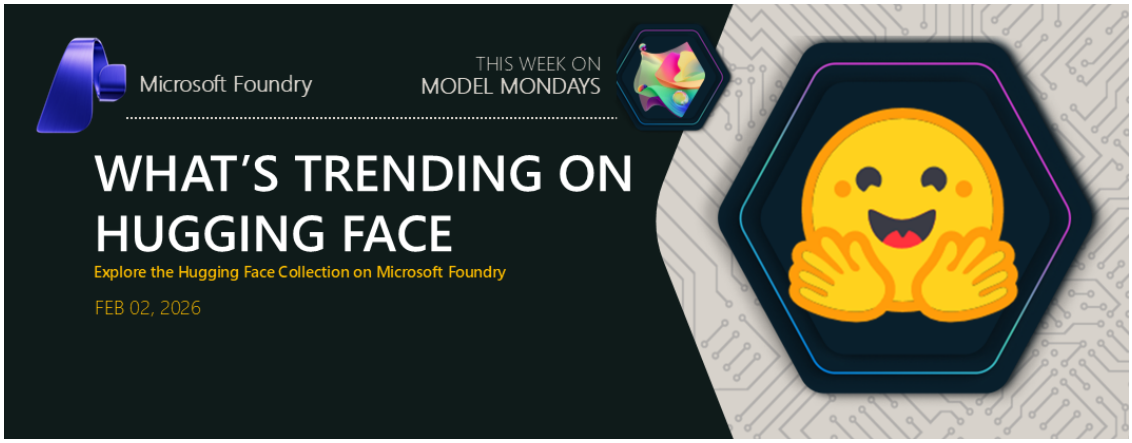
What's trending in Hugging Face? Feb 2, 2026

Each week we showcase trending Hugging Face models that are now available in Microsoft Foundry.



Open-source AI is moving fast, with important breakthroughs in reasoning, agentic systems, multimodality, and efficiency emerging every day. Hugging Face has been a leading platform where researchers, startups, and developers share and discover new models. Microsoft Foundry brings these trending Hugging Face models into a production-ready experience, where developers can explore, evaluate, and deploy them within their Azure environment. Our weekly Model Monday's series highlights Hugging Face models available in Foundry, focusing on what matters most to developers: why a model is interesting, where it fits, and how to put it to work quickly.

This week's Model Mondays edition highlights three Hugging Face models, including a powerful Mixture-of-Experts model from Z. AI designed for lightweight deployment, Meta's unified foundation model for image and video segmentation, and MiniMax's latest open-source agentic model optimized for complex workflows.

Z.AI's GLM-4.7-flash

Model Basics

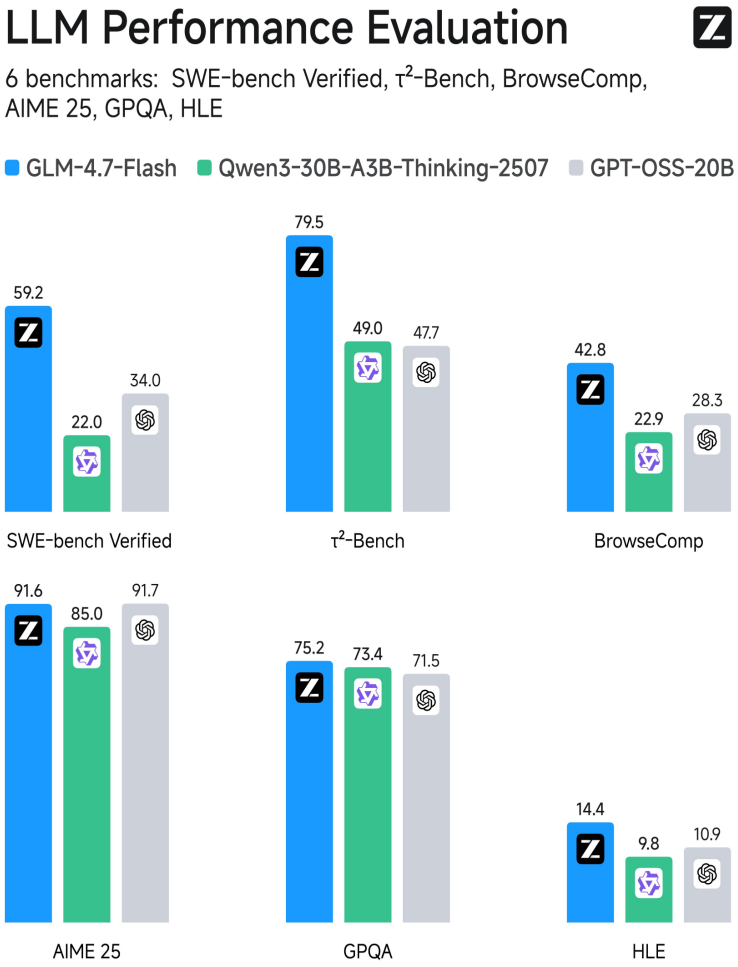
- **Model name:** zai-org/GLM-4.7-Flash
- **Parameters / size:** 30B-3B (active)
- **Default settings**: 131,072 max new tokens

- **Primary task:** Agentic, Reasoning and Coding

Why this model matters

- **Why it's interesting:**
  - It utilizes a Mixture-of-Experts (MoE) architecture (30B total parameters and 3B active parameters) to offer a new option for lightweight deployment.
  - It demonstrates strong performance on logic and reasoning benchmarks, outperforming similar sized models like gpt-oss-20b on AIME 25 and GPQA benchmarks.
  - It supports advanced inference features like "Preserved Thinking" mode for multi-turn agentic tasks.
- **Best-fit use cases:**
  - Lightweight local deployment, multi-turn agentic tasks, and logical reasoning applications.
- **What's notable:**
  - From the Foundry catalog, users can deploy on a A100 instance or unsloth/GLM-4.7-Flash-GGUF on a CPU.

Evidence



Try it

Use case	Best practice prompt pattern-practice prompt pattern
Agentic coding (multi-step repo work, debugging, refactoring)	Treat the model as an autonomous coding agent, not a snippet generator. Explicitly require task decomposition and step-by-step execution, then a single consolidated result.
Long-context agent workflows (local or low-cost autonomous agents)	Call out long-horizon consistency and context preservation. Instruct the model to retain earlier assumptions and decisions across turns.

Now that you know GLM-4.7-Flash works best when you give it a clear goal and let it reason through a bounded task, here's an example prompt that a product or engineering team might use to identify risks and propose mitigations:

> You are a software reliability analyst for a mid-scale SaaS platform.
>
> Review recent incident reports, production logs, and customer issues to uncover edge-case failures outside normal usage (e.g., rare inputs, boundary conditions, timing/concurrency issues, config drift, or unexpected feature interactions).
>
> Prioritize low-frequency, high-impact risks that standard testing misses. Recommend minimal, low-cost fixes (validation, guardrails, fallback logic, or documentation).
>
> Deliver a concise executive summary with sections: Observed Edge Cases, Root Causes, User Impact, Recommended Lightweight Fixes, and Validation Steps.

Meta's Segment Anything 3 (SAM3)

Model Basics

- **Model name:** facebook/sam3
- **Parameters / size:** 0.9B
- **Primary task:** Mask Generation, Promptable Concept Segmentation (PCS)

Why this model matters

- **Why it's interesting:**
  - It handles a vastly larger set of open-vocabulary prompts than SAM 2, and unifies image and video segmentation capabilities.

    - It includes a "SAM 3 Tracker" mode that acts as a drop-in replacement for SAM 2 workflows with improved performance.
- **Best-fit use cases:**
    - Open-vocabulary object detection, video object tracking, and automatic mask generation
- **What's notable:**
    - Introduces Promptable Concept Segmentation (PCS), allowing users to find all matching objects (e.g., "dial") via text prompt rather than just single instances.

Try it

This model enables users to identify specific objects within video footage and isolate them over extended periods. With just one line of code, it is possible to detect multiple similar objects simultaneously. The accompanying GIF demonstrates how SAM3 efficiently highlights players wearing white on the field as they appear and disappear from view. Additional examples are available at the following repository: https://github.com/facebookresearch/sam3/blob/main/assets/player.gif

Use case	Bestpractice prompt pattern-practice prompt pattern
Agentic coding (multi-step repo work, debugging, refactoring)	Treat SAM 3 as a *concept detector*, not an interactive click tool. Use short, concrete noun-phrase concept prompts instead of describing the scene or asking questions. Example prompt: “yellow school bus” or “shipping containers”. Avoid verbs or full sentences.
Video segmentation + object tracking	Specify the same concept prompt once, then apply it across the video sequence. Do not restate the prompt per frame. Let the model maintain identity continuity. Example: “person wearing a red jersey”.
Hard-to-name or visually subtle objects	Use exemplar-based prompts (image region or box) when text alone is ambiguous. Optionally combine positive and negative exemplars to refine the concept. Avoid over-constraining with long descriptions.

Using the GIF above as a leading example, here is a prompt that shows how SAM 3 turns raw sports footage into structured, reusable data. By identifying and tracking players based on visual concepts like jersey color so that sports leagues can turn tracked data into interactive experiences where automated player identification can relay stats, fun facts, etc when built into a larger application. Here is a prompt that will allow you to start identifying specific players across video:

> Act as a sports analytics operator analyzing football match footage. Segment and track all football players wearing blue jerseys across the video. Generate pixel-accurate segmentation masks for each player and assign persistent instance IDs that remain stable during camera movement, zoom, and player occlusion. Exclude referees, opposing team jerseys, sidelines, and crowd. Output frame-level

masks and tracking metadata suitable for overlays, player statistics, and downstream analytics pipelines.

MiniMax AI's MiniMax-M2.1

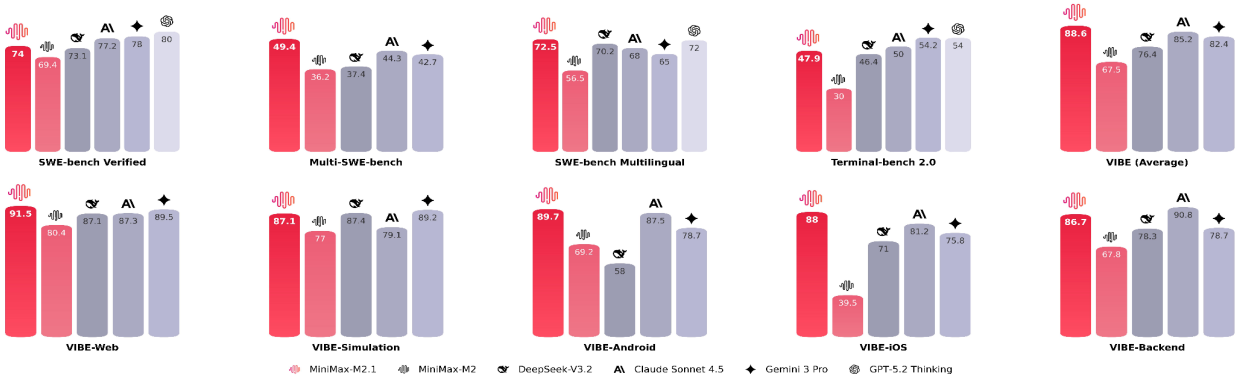
Model Basics

- **Model name:** MiniMaxAI/MiniMax-M2.1
- **Parameters / size:** 229B-10B Active
- **Default settings**: 200,000 max new tokens
- **Primary task:** Agentic and Coding

Why this model matters

- **Why it's interesting:**
  - It is optimized for robustness in coding, tool use, and long-horizon planning, outperforming Claude Sonnet 4.5 in multilingual scenarios.
  - It excels in full-stack application development, capable of architecting apps "from zero to one".
  - The model delivers exceptional stability across various coding agent frameworks.
- **Best-fit use cases:**
  - Lightweight local deployment, multi-turn agentic tasks, and logical reasoning applications.
- **What's notable:**
  - The release of open-source weights for M2.1 delivers a massive leap over M2 on software engineering leaderboards.

Evidence


74
69.4
73.1
77.2
78
80
SWE-bench Verified
49.4
36.2
37.4
44.3
42.7
Multi-SWE-bench
72.5
56.5
70.2
68
65
72
SWE-bench Multilingual
47.9
30
46.4
50
54.2
54
Terminal-bench 2.0
88.6
67.5
76.4
85.2
82.4
VIBE (Average)
91.5
80.4
87.1
87.3
89.5
VIBE-Web
87.1
77
87.4
79.1
89.2
VIBE-Simulation
89.7
69.2
58
87.5
78.7
VIBE-Android
88
39.5
71
81.2
75.8
VIBE-iOS
86.7
67.8
78.3
90.8
78.7
VIBE-Backend
MiniMax-M2.1
MiniMax-M2
DeepSeek-V3.2
Claude Sonnet 4.5
Gemini 3 Pro
GPT-5.2 Thinking

Try it

Use case	Bestpractice prompt pattern-practice prompt pattern
End-to-end agentic coding (multi-file edits, run-fix loops)	Treat the model as an autonomous coding agent, not a snippet generator. Explicitly require task decomposition and step-by-step execution, then a single consolidated result.
Long-horizon tool-using agents (shell, browser, Python)	Explicitly request stepwise planning and sequential tool use. MiniMax-M2 is built to plan and execute long toolchains while maintaining state; prompts should emphasize task completion, not conversation.
Long-context reasoning & analysis (large documents / logs)	Declare the scope and desired output structure up front. MiniMax-M2 performs best when the objective and final artifact are clear, allowing it to manage long context and maintain coherence.

Because MiniMax-M2 is designed to act as a long-horizon analytical agent, it shines when you give it a clear end goal and let it work through large volumes of information—here's a prompt a risk or compliance team could use in practice:

> You are a financial risk analysis agent. Analyze the following transaction logs and compliance policy documents to identify potential regulatory violations and systemic risk patterns. Plan your approach before executing. Work through the data step by step, referencing evidence where relevant. Deliver a final report with the following sections: Key Risk Patterns Identified, Supporting Evidence, Potential Regulatory Impact, Recommended Mitigations. Your response should be a complete, executive-ready report, not a conversational draft.

Getting started

You can deploy open-source Hugging Face models directly in Microsoft Foundry by browsing the Hugging Face collection in the Foundry model catalog and deploying to managed endpoints in just a few clicks. You can also start from the Hugging Face Hub—select any supported model and choose Deploy on Microsoft Foundry, which brings you straight into Azure with secure, scalable inference already configured. Learn how to discover models and deploy them using Microsoft Foundry documentation.

- **Follow along the Model Mondays series and access the GitHub to stay up to date on the latest**
- **Read Hugging Face on Azure docs**
- *Learn about One-click deployments from the Hugging Face Hub on Microsoft Foundry*
- **Explore models in Microsoft Foundry**