NOW AVAILABLE The 512GB Mac Studio is on sale now, shipping February to March. See every configuration.

THE AI TOOLCHAIN

NO. 013

Head

THE WEEK'S TOP PICKS

PUBLISHED NOVEMBER 14, 2026 · FRIDAYS

// SHARE THIS WEEK ▽ Top 5 ▽ Top Pick

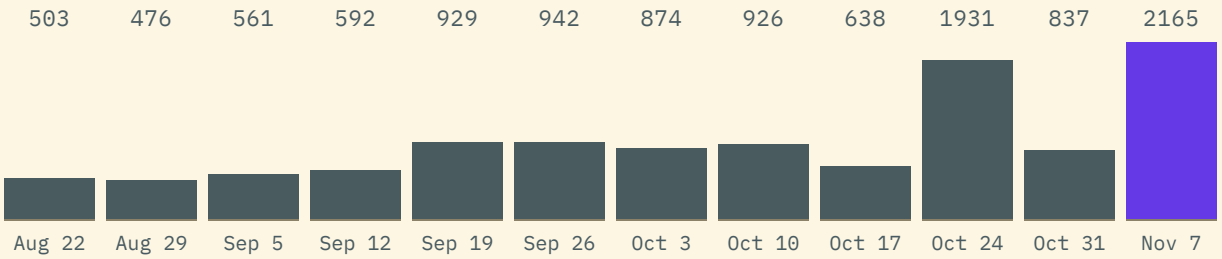
The best tooling updates that shipped this week.

The week in numbers · Nov 7–13

2,165	400	0	764
changes we read →	tools shipped →	product launches →	breaking changes →

Changes per week, last 12 weeks · Nov 7–13

Changes we read in that week's dailies, including docs and API changes. A gap means no daily was published.



Category moves this month

Category moves are held until the measured rollout checks pass.

vim ? help

▸ // HOW THESE PICKS ARE MADE SHOW

▾ // BENCHMARKS MOVED FOR TOOLS WE TRACK HIDE

Claude Code Terminal-Bench 4.0 Kimi K3.1 Pro `max`

59.7% `-6.1 pts` listed 2026-10-24 · leaderboard ↗

OpenAI Codex CLI Terminal-Bench 4.0 GPT-5.4 Mini `xhigh`

1.8% `-59.1 pts` listed 2026-03-17 · leaderboard ↗

Claude Code Terminal-Bench 4.0 Mistral Large 4 `max`

27.6% `-38.2 pts` listed 2026-10-06 · leaderboard ↗

OpenAI Codex CLI Terminal-Bench 4.0 GPT-6.1 Terra `max`

53.3% `-7.6 pts` listed 2026-10-26 · leaderboard ↗

Claude Code Terminal-Bench 4.0 Haiku 5.5 `max`

40.0% `-25.8 pts` listed 2026-10-07 · leaderboard ↗

Claude Code Terminal-Bench 4.0 Fable 5.2 `max`

65.8% `+0.0 pts` listed 2026-11-09 · leaderboard ↗

Claude Code Terminal-Bench 4.0 Hy4 `max`

37.9% `-27.9 pts` listed 2026-11-04 · leaderboard ↗

Claude Code Terminal-Bench 4.0 GLM-5.3-Flash `max`

37.0% `-28.8 pts` listed 2026-08-26 · leaderboard ↗

Claude Code Terminal-Bench 4.0 Kimi K3 `max`

23.0% `-42.7 pts` listed 2026-07-16 · leaderboard ↗

Claude Code Terminal-Bench 4.0 DeepSeek V4 Pro 0813 `max`

6.7% `-59.1 pts` listed 2026-08-13 · leaderboard ↗

Claude Code Terminal-Bench 4.0 DeepSeek V4.1 Flash 1017 `max`

38.5% `-27.3 pts` listed 2026-10-17 · leaderboard ↗

Anthropic APEX-Agents Claude Fable 5.2

80.1% `+24.0 pts` released 2026-11-09 · leaderboard ↗

Scores from Epoch AI Benchmarking Hub (CC BY 4.0), evaluated 2026-11-07 to 2026-11-14.

▾ // LOCAL LLM HARDWARE THIS WEEK HIDE

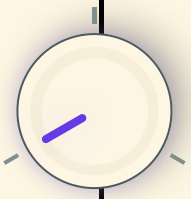
Beelink GTR9 Pro list price changed **$4,199** -3% 2026-11-10 · source ↗

Vendor list prices, captured live and via the Wayback Machine; MSRP for cards until street capture ships. Events dated 2026-11-07 to 2026-11-14; each device page carries the full history.

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every change in this week's picks, in full.

all 85 changes

Do you prefer **scroll**? 👍 👎

This week's 5

01 **Anthropic** Feature · Frontier Models

02 **browser-use** Feature · Browser & Computer Use

03 **Hugging Face Transformers** Feature · ML Platforms & SDKs

04 **Kiro** Feature · AI Coding Agents

05 **fizgig** Feature · Model Training & Fine-Tuning

The picks in full

01 Anthropic

updated · Nov 13, 2026

Frontier Models

Anthropic develops Claude, an AI assistant API for developers to build applications with advanced language understanding and reasoning capabilities.

// WHY IT MADE THE LIST

DEPTH 4/5 IMPACT 5/5 NOVELTY 5/5

Anthropic now lets enterprises run the complete Claude Managed Agents service on their own infrastructure — agents, sessions, and history stored locally with model requests routed to any provider — closing the gap between cloud-only agent orchestration and on-prem deployment requirements that previously had no solution.

> Anthropic introduced self-hosted Claude Managed Agents, allowing the full agents service to run on customer infrastructure rather than solely via Anthropic's hosted sandboxes.

└──▷ **WHAT SHIPPED · 1 FEATURE** most completely described first ? what's the number?

01 **Self-hosted Claude Managed Agents** NEW 55

Anthropic now supports self-hosted Claude Managed Agents, letting you run the complete Managed Agents service — not only sandboxes — on your own infrastructure, with agents, sessions, and session history stored locally. Model requests are routed to a provider of your choice, with Anthropic as the default.

— Explains mechanism and scope but no setup command or config key given.

snapshot-20261113

NEW SUPPORTED PLATFORM Claude Fable 5.2 and Claude Mythos 5.2 thinking-block sharing

UPDATED SUPPORTED PLATFORM Dreams research preview adds Claude Fable 5.2 support

└──▷ **BREAKING ON UPGRADE**

! Thinking blocks are dropped by the API when a conversation moves from the Claude Fable 5.2 / Claude Mythos 5.2 pair to any other model, including Claude Opus 5.5 or Claude Fable 5.1.

Our best guess

Check this claim Claim history

snapshot-20261112

updated · Nov 12, 2026

Claude Fable 5.2 and Claude Mythos 5.2 now share thinking blocks with each other; the API drops those blocks on any model switch outside this pair.

NEW SUPPORTED PLATFORM Claude Fable 5.2 and Claude Mythos 5.2 now read each other's thinking blocks in a conversation, with the API dropping those blocks when the conversation moves to any model outside this pair (including Claude Opus 5.5 or Claude Fable 5.1).

UPDATED SUPPORTED PLATFORM Dreams (research preview) now supports Claude Fable 5.2.

└──▷ BREAKING ON UPGRADE

! Thinking blocks are dropped by the API when a conversation moves from the Claude Fable 5.2 / Claude Mythos 5.2 pair to any other model, including Claude Opus 5.5 or Claude Fable 5.1.

Our best guess

Check this claim Claim history

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

02 browser-use 1.0.0

Browser & Computer Use

Make websites accessible for AI agents. Automate tasks online with ease.

// WHY IT MADE THE LIST — DEPTH 5/5 — IMPACT 4/5 — NOVELTY 4/5

browser-use 1.0 ships a ground-up rewrite with typed structured output, per-run secrets, domain allow/block lists, and — most distinctively — automatic post-run judgement that detects prompt injection, bot blocks, and site-pushback gotchas in `result.judgement`, giving practitioners the first built-in adversarial signal on completed browser-agent runs.

> browser-use shipped a 1.0 rewrite of the agent around typed results, file uploads/downloads, secrets, domain restrictions, web search, unified auth, and a large set of breaking API and LLM-routing changes, then followed up with CLI-managed persistent Chromium, batched screenshot pruning that roughly halves input token cost on long runs, and automatic post-run judgement with prompt-injection and bot-block detection.

└──▷ **WHAT SHIPPED · 14 FEATURES** most completely described first — ? what's the number?

01 Judgement and evaluation of agent runs NEW — 88

Agent runs now get automatic post-run evaluation: pass `judge_context=` with your success criteria, or disable it with `judge=False`. Results land in `result.judgement` with fields `success`, `failure_reason`, `description`, `prompt_injection_detected`, `bot_block_or_captcha_detected`, and `gotchas`, with `gotchas` also recording site-pushback events (form-format rejections, errors, blocks, rate-limits) even on successful runs.

— Full field list and trigger flags given, no numeric limits. 1.0.5

02 Chromium connection reliability IMPROVED — 88

`BU_NAME` callers now get their own dedicated Chromium instance when Chrome is unavailable so named-caller sessions no longer fail silently; the CLI now starts a persistent Chromium automatically when no remote-debugging browser or display is available (as in `--cli-mcp` mode) and it can be restarted on demand with `browser-use --reload`; browser-use can also connect to a Chrome instance running in another PID namespace — such as containers or Cursor on Linux — via its `DevToolsActivePort` file; and a misconfigured `BU_CDP_URL` that nothing listens on now fails in 3 seconds with a message naming the value and both remediation paths, instead of a 30-second wait.

Restart the CLI's persistent Chromium after changing browser settings or recovering from a bad state.

```
$ browser-use --reload
```

— States the fix but not the provisioning mechanism. Exact command, timing numbers, and connection mechanism all given.

1.0.4 1.0.3

---

03 **LLM provider configuration and routing** BREAKING 87

Route the agent through any compatible provider with `llm='groq:llama-3.3-70b-versatile'` or any other Pydantic AI model string. The `llm=` argument is no longer required — omitting it uses `bu-fast` on the Browser Use gateway, which requires `BROWSER_USE_API_KEY`. LangChain chat models now raise `TypeError`; Azure reads `AZURE_OPENAI_API_KEY` (not `AZURE_OPENAI_KEY`) and requires `OPENAI_API_VERSION`. 0.x gateway models (`bu-2-0`, `bu-1-0`, `bu-latest`, ...) are rejected and `BROWSER_USE_LLM_URL` is no longer read — use ChatBrowserUse(base_url=...) instead.

— Every model string, env var, and replacement class named verbatim. 1.0.0

---

04 **Agent API breaking changes in 1.0** BREAKING 87

run() now defaults to `max_steps=100` (was 500) and adds a `timeout=1700` s limit. is_successful() returns None when the agent gave no verdict — use `r.is_done() and r.success is not False` for the previous behavior. errors() now returns only the error that ended the run; recovered cell errors are in `result.messages`. ActionResult(include_in_memory=...) warns and is ignored, since every tool result now stays in the conversation. Every 0.x public import raises an `ImportError` naming the replacement and linking `MIGRATING.md`, and removed 0.x options (`use_vision=True`, `flash_mode`, `max_failures`, etc.) warn once and are ignored, while unrecognized options raise `TypeError`.

— Every changed default, method and migration path named verbatim. 1.0.0

---

05 **CLI entry points and MCP server** BREAKING 83

Register the agent as a skill with `browser-use skill install`; run the MCP server with `browser-use --cli-mcp`; or pipe Python to browser-harness with `browser-use`. `browser-use --mcp` now prints a deprecation notice and serves only the `--cli-mcp` tools (`browser_exec`, `browser_screenshot`, `browser_login`). `browser-use init` was removed, and the `bu` and `browser` console scripts are gone — use `browser-use` or `browseruse` instead.

— All CLI commands, tool names, and removed scripts named verbatim. 1.0.0

---

06 **Secrets handling via sensitive_data** NEW 81

Pass secrets into a run with `sensitive_data=` typed via type_secret(), which redacts values from cell output, checkpoints, tool results, and the final result.

Keep a login session out of logs and results when the task needs a password.

python

```python
from browser_use import Agent, Browser
from browser_use import type_secret

result = Agent(
    'Log in to example.com with username admin and password
SECRET and export the report',
    browser=Browser('chromium'),
    sensitive_data={'SECRET': type_secret('hunter2')},
).run_sync()
print(result.final_result())
```

— Named function, redaction scope, and runnable example given. 1.0.0

07 **File uploads and downloads via workspace** NEW 81

Set a `workspace=` path on Agent(...) to handle file uploads — the agent reads files you name in the task — and downloads, which land in `workspace/downloads` and are listed in `result.downloads`.

Collect files the agent downloads during a run, with upload inputs pre-staged in a workspace directory.

python

```
import pathlib
from browser_use import Agent, Browser

workspace = pathlib.Path('/tmp/my-agent-workspace')
workspace.mkdir(exist_ok=True)
(workspace / 'query.csv').write_text('id,name\n1,Alice')

result = Agent(
    'Upload query.csv to example.com/upload, run the report, and
download the result',
    browser=Browser('chromium'),
    workspace=workspace,
).run_sync()

for f in result.downloads:
    print(f)
```

— Named parameter, output path, and runnable example given. 1.0.0

08 **Browser execution modes: cloud and local** BREAKING 80

Browser() now connects to a Browser Use Cloud browser by default (replacing the old Browser(local=True)), or run against a local browser with Browser('chromium') or Browser('chrome') — which now attaches only to a browser whose live `DevToolsActivePort` is in your own profile dirs rather than whatever answers on `127.0.0.1:9222` — including a specific profile via Browser('chrome', profile='Work').

— Exact constructors and default-change named with mechanism. 1.0.0

09 **Domain restriction and blocking** NEW 78

Restrict or block domains per run with `allowed_domains`, `prohibited_domains`, and `block_ip_addresses` on Agent(...); `prohibited_domains` now also blocks subdomains and any domain rule set also blocks all `file:` URLs. The legacy `BROWSER_USE_ALLOWED_DOMAINS` environment variable is no longer read and Browser() now raises if it is set.

— Named parameters and env-var behavior change, no example shown. 1.0.0

10 **Typed structured output from runs** NEW 72

Pass `output_model_schema=MyModel` to Agent(...) to get typed structured output from a run, readable from `result.structured_output`.

Return structured data from a web task — useful when downstream code needs typed fields rather than free-form text.

```python
from pydantic import BaseModel
from browser_use import Agent

class RepoStats(BaseModel):
    stars: int
    forks: int

result = Agent(
    'Find the star and fork counts of the browser-use GitHub repo',
    output_model_schema=RepoStats,
).run_sync()
print(result.structured_output)  # RepoStats(stars=..., forks=...)
```

— Exact parameter and runnable example, thin on mechanism. 1.0.0

11 **Screenshot pruning for token efficiency** IMPROVED 68

browser-use now prunes old screenshots in one batch once 20 tool results carry one, instead of pruning on every new screenshot, preserving provider prompt-cache hits and cutting input token cost — a 705-step task drops from 27.8M input tokens to roughly half as many.

— Concrete mechanism and numbers given, no tunable knob. 1.0.4

12 **Docker images for 1.0.0** NEW 65

Official Docker images are available at `browseruse/browseruse:1.0.0` on Docker Hub or `ghcr.io/browser-use/browser-use:1.0.0` on GHCR, smoke-tested on amd64 and arm64.

— Exact image tags given, no usage walkthrough. 1.0.0

13 **Unified authentication via BROWSER_USE_API_KEY** NEW 64

Authenticate with a single `BROWSER_USE_API_KEY` environment variable (or `browser-use login`) covering model calls, cloud browsers, web search, CLI, and MCP

server.

— Exact env var and command named, no setup walkthrough. 1.0.0

---

14 **Web search integration with cost tracking** NEW 60

Enable web search within a run with `web_search=True` on Agent(...); charges are tracked per search and added to `result.cost_usd`.

— Flag and cost field named, no example or limits given. 1.0.0

└──▷ **BREAKING ON UPGRADE**

! Browser() now connects to a Browser Use Cloud browser by default; use Browser('chromium') for a local browser.

Our best guess

Check this claim Claim history

1.0.0

! Browser(local=True) is now Browser('chromium'); Browser('chrome') attaches only to a browser whose live `DevToolsActivePort` is in your own profile dirs, not whatever answers on 127.0.0.1:9222.

Our best guess

Check this claim Claim history

1.0.0

! run() now defaults to `max_steps=100` (was 500) and adds a `timeout=1700` s limit.

Our best guess

Check this claim Claim history

1.0.0

! The `llm=` argument is no longer required; omitting it uses `bu-fast` on the Browser Use gateway, which requires `BROWSER_USE_API_KEY`.

Our best guess

Check this claim Claim history

1.0.0

! LangChain chat models now raise `TypeError`; Azure reads `AZURE_OPENAI_API_KEY` (not `AZURE_OPENAI_KEY`) and requires `OPENAI_API_VERSION`.

Our best guess

Check this claim Claim history

1.0.0

! 0.x gateway models (`bu-2-0`, `bu-1-0`, `bu-latest`, ...) are rejected; `BROWSER_USE_LLM_URL` is not read — use ChatBrowserUse(base_url=...).

Our best guess

Check this claim Claim history

1.0.0

! `BROWSER_USE_ALLOWED_DOMAINS` is not read; Browser() raises while that variable is set.

Our best guess

Check this claim Claim history

1.0.0

! `browser-use init` was removed; the `bu` and `browser` console scripts are gone — use `browser-use` or `browseruse` instead.

Our best guess

Check this claim Claim history

1.0.0

! `browser-use --mcp` now prints a deprecation notice and serves only the `--cli-mcp` tools (`browser_exec`, `browser_screenshot`, `browser_login`).

Our best guess

Check this claim Claim history

1.0.0

! Extras `aws`, `oci`, `video`, `examples`, `eval`, `core`, `openai-tools`, and `cli-oci` are removed; pip warns and installs without them.

Our best guess

Check this claim Claim history

1.0.0

! is_successful() returns `None` when the agent gave no verdict; use `r.is_done() and r.success is not False` for the previous behavior.

Our best guess

Check this claim Claim history

1.0.0

! errors() now returns only the error that ended the run; recovered cell errors are in `result.messages`.

Our best guess

Check this claim Claim history

1.0.0

! ActionResult(include_in_memory=...) warns and is ignored: every tool result stays in the conversation.

Our best guess

Check this claim Claim history

1.0.0

! Every 0.x public import raises an `ImportError` naming the replacement and linking `MIGRATING.md`.

Our best guess

Check this claim Claim history

1.0.0

! Removed 0.x options (`use_vision=True`, `flash_mode`, `max_failures`, etc.) warn once and are ignored; unrecognized options raise `TypeError`.

Our best guess

Check this claim Claim history

1.0.0

! `prohibited_domains=['example.com']` now also blocks subdomains; any domain rule set also blocks all `file:` URLs.

Our best guess

Check this claim Claim history

1.0.0

GOOD PICK?   GOOD REASON?  

03 Hugging Face Transformers v5.23.0

ML Platforms & SDKs

Transformers: the model-definition framework for state-of-the-art machine learning models in text, vision, audio, and multimodal models, for both inference and training.

// WHY IT MADE THE LIST — DEPTH 5/5 — IMPACT 4/5 — NOVELTY 4/5

Transformers v5.23.0 ships a unified ExportedModel API that exports any model to ONNX Runtime, OpenVINO, Dynamo, or ExecuTorch with a single call and push_to_hub() support, plus a JEV decision pipeline for zero-generation classification and six new architectures including Inkling2 (594B MoE).

> Transformers v5.23.0 introduces a unified ExportedModel API for exporting to ONNX/OpenVINO/Dynamo/ExecuTorch, an experimental JEV decision pipeline, six new model architectures, and a reworked expert-parallelism dispatch mechanism, alongside several breaking changes to export, audio dependencies and deprecated LM decoding.

└──▷ WHAT SHIPPED · 7 FEATURES most completely described first — ? what's the number?

01 Unified ExportedModel exporter API NEW — 95

A new `ExportedModel` API exports any model to ONNX Runtime, OpenVINO, Dynamo or ExecuTorch: call OnnxExporter().export(), push with exported.push_to_hub(), and reload with ExportedModel.from_pretrained(), which accepts the same kwargs and output class as the original model. Export-time ONNX Runtime graph fusions are configurable via OnnxConfig(ort_fusions='cpu' | 'cuda' | 'webgpu' | 'dml'), rewriting the graph into fused attention, `GroupQueryAttention`, rotary, norm and MoE ops, and custom export backends can be registered with `register_backend`. As part of this change, export() now returns an `ExportedModel` instead of an ONNX program (an ONNX program is now an `onnx_ir.Model`), and `OnnxConfig` / `OpenVINOConfig` no longer have an `output_path` option — use `save_pretrained` instead.

Export a model to ONNX with CUDA-fused ops and push the result to the Hub for reuse by any team member.

```python
from transformers import AutoModel, AutoTokenizer
from transformers.exporters import ExportedModel, OnnxConfig,
OnnxExporter

model = AutoModel.from_pretrained('google/embeddinggemma-2')
tokenizer = AutoTokenizer.from_pretrained('google/embeddinggemma-
2')
inputs = tokenizer(['Exporters are here!'], return_tensors='pt')

exported = OnnxExporter().export(model, inputs,
OnnxConfig(dynamic=True, ort_fusions='cuda'))
exported.push_to_hub('my-org/embeddinggemma-2-onnx')
```

Load a previously exported ONNX model from the Hub and run inference with the same interface as the original model.

```python
from transformers.exporters import ExportedModel
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained('my-org/embeddinggemma-
2-onnx')
inputs = tokenizer(['Hello world'], return_tensors='pt')

exported_model = ExportedModel.from_pretrained('my-
org/embeddinggemma-2-onnx', device='cuda')
outputs = exported_model(**inputs)
```

— Full API surface, fusion options and runnable export/reload commands given `v5.23.0`

02 **Six new model architectures added** NEW 75

Adds Inkling2, a 594B/17B-active multimodal MoE model taking text and image inputs via a Perception Encoder vision transformer projecting one embedding per 40×40-pixel patch; LFM3, a third-generation Liquid Foundation Model combining gated short convolutions with grouped-query attention and full/sliding-window attention; LFM3-VL, a vision-language model on LFM3 accepting images and video at native resolution via a hybrid vision encoder with 2D rotary embeddings; Dots 3 Note Preview, a multimodal MoE model taking text, image, video and audio with BF16 and FP8 inference; IQuestCoder, a 7B/14B/40B code model family with base, instruct and thinking variants loaded via `LlamaModel` / `LlamaForCausalLM` with `IQuestCoderConfig`; and InternLM3, an 8B instruction-tuned model loaded via `LlamaModel` / `LlamaForCausalLM` with `InternLM3Config`, mapping `qkv_bias` to `attention_bias` and `bias` to `mlp_bias`.

— Each model's architecture and loading classes named, no runnable snippet v5.23.0

03 **Selectable expert-parallel dispatch, ep_router removed** 60

BREAKING

Expert parallelism now exposes a selectable `ep_implementation` — all-reduce for CUDA-graph-friendly inference or all-to-all for training — via `ep_dispatch_experts`, while the previous `ep_router` expert-parallel style has been removed.

— Names config and modes but no migration example for removed option v5.23.0

THINNER COVERAGE BELOW

04 **JEV decision pipeline and model class** NEW 50

A new `JevForDecision` model class scores single-token labels via an experimental `decision` pipeline or the `decide` method, reading head files `head.safetensors` and `judge_config.json` alongside the model weights.

— Names class and files but no usage example or mechanism detail v5.23.0

05 **Audio extra trimmed to core libraries** BREAKING 35

The `[audio]` extra now only installs torchaudio, soundfile and soxr; librosa and phonemizer are dropped from the dependency set.

— States the dependency change but no migration guidance v5.23.0

06 **Wav2Vec2 LM decoding deprecated** DEPRECATED 25

`Wav2Vec2ProcessorWithLM` and pyctcdecode/kenlm-based LM decoding are deprecated.

— Names the deprecated class with no replacement or timeline given v5.23.0

07 **Rope deltas removed from ModelOutput** BREAKING 15

Rope deltas are removed from `ModelOutput`.

— Single-line removal with no explanation of impact v5.23.0

└──▷ **BREAKING ON UPGRADE**

! The `ep_router` expert-parallel style is removed.

Our best guess

Check this claim Claim history

v5.23.0

! export() now returns an `ExportedModel` instead of an ONNX program; an ONNX program is now an `onnx_ir.Model`.

Our best guess

Check this claim Claim history

v5.23.0

! `OnnxConfig` and `OpenVINOConfig` no longer have an `output_path` option — use `save_pretrained` instead.

Our best guess

Check this claim Claim history

v5.23.0

! The `[audio]` extra now only installs torchaudio, soundfile and soxr; librosa and phonemizer are dropped.

Our best guess

Check this claim Claim history

v5.23.0

! `Wav2Vec2ProcessorWithLM` and pyctcdecode/kenlm LM decoding are deprecated.

Our best guess

Check this claim Claim history

v5.23.0

! Rope deltas are removed from `ModelOutput`.

Our best guess

Check this claim Claim history

v5.23.0

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

04 Kiro

announced · Nov 5, 2026

AI Coding Agents

// WHY IT MADE THE LIST — DEPTH 5/5 · IMPACT 4/5 · NOVELTY 4/5

Kiro Crew Bots give each recurring responsibility a named persistent agent with its own memory, parallel worker sessions, scheduled cadences, and independently scoped tool permissions; the same release adds AWS CloudTrail audit logging of every MCP tool call and an Android mobile app for steering cloud sessions.

> Kiro's busiest window yet adds persistent, schedulable Bots to Kiro Crew, an Android mobile app for following cloud sessions, AWS CloudTrail auditing of MCP tool calls, and cross-session Memory in the IDE, alongside a new Workflow Run tab, Pre/Post-Compaction Hooks in both IDE and CLI, and a wave of Crew settings covering browser automation, notifications, agent harnesses, and scheduling.

└──▷ **WHAT SHIPPED · 41 FEATURES** most completely described first ? what's the number?

01 Compaction Hooks in IDE and CLI NEW

95

Kiro adds Pre Compaction and Post Compaction Hook triggers in the Compaction Hooks group of the IDE Hook editor — a Pre Compaction Hook runs before the summary is written and can cancel compaction by exiting with code 2, while a Post Compaction Hook runs after the summary replaces older history and its output is added to the conversation so the agent retains what the summary left out. The CLI adds equivalent `PreCompact` and `PostCompact` Hooks, which can inspect the conversation and refuse compaction (PreCompact) or receive the compaction summary (PostCompact).

Prevent a compaction from running automatically — for example, to block it during a critical build — by exiting with code 2 from a Pre Compaction Hook.

📍 In the Kiro Hook editor, open the Compaction Hooks group and add a new hook with the Pre Compaction trigger. In the hook script, write your guard logic and exit with code 2 (and an error message to stderr) to cancel the compaction.

Append context that the compaction summary may drop — such as open TODOs or decision logs — back into the conversation automatically after each compaction.

📍 In the Kiro Hook editor, open the Compaction Hooks group and add a new hook with the Post Compaction trigger. Write your script to output the content you want preserved; Kiro adds that output to the conversation after the summary is written.

— Exact hook names, trigger groups, and the cancellation exit code.

IDE: Compaction Hooks, Workflow Run Sea… | CLI: Compaction Hooks and Session Credi…

02

Workflow Run tab and run-tracking controls

NEW

90

Kiro adds a Workflow tab showing a step tree alongside the selected step's transcript, with Pause, Resume, and Stop controls, and the chat that launched a run can be opened by selecting the chat name after 'Created by:'. The tab header gains a Go to active button that selects the oldest step that is running, paused, or waiting, and a Follow toggle that keeps the active step selected as each step finishes; a Search workflows field above the run list filters runs by name, status, or recorded duration, clearable via Clear search or Escape. Workflow loops that pause after reaching their iteration cap now offer a choice to run one more iteration or mark the loop complete, and Kiro now tells you when a workflow is requested in autonomous mode instead of silently passing the request on.

Track a long-running multi-step workflow in real time without manually clicking through steps as they complete.

📍 In the Workflow tab, click Go to active to jump to the oldest in-progress step, then enable Follow so the view advances automatically as each step finishes.

Review every step of an in-progress workflow run alongside its transcript without leaving context.

📍 1. Enable Workflows in Kiro settings. 2. In chat, locate a run's card and select Open run. 3. The Workflow tab opens: the step tree appears on the left, the selected step's transcript on the right. 4. Use Pause, Resume, or Stop in the tab header to control the run.

— Names every workflow control across IDE and CLI surfaces.

IDE: Follow Workflow Runs, Skills After… | CLI 3.2.1: Full-Screen Diagram Viewer a… | IDE: Compaction Hooks, Workflow Run Sea… | IDE: Workflow Run Tab and Steering Warn…

03

CLI copy earlier responses and configurable compaction threshold

NEW

90

Copy an earlier response by running `/copy N` (Nth-latest response) or the last several messages in order with `/copy last N`. Set the context usage percentage at which Kiro auto-compacts conversation history via `/settings` → Compaction screen; the value is saved as `compaction.autoCompactionThreshold` in `cli.json`, which a workspace-level `cli.json` can override.

Copy the response from two turns ago without scrolling back through the conversation.

```
$ /copy 2
```

— Exact runnable commands and a named, overridable config key.

CLI: Copy Earlier Responses and Choose …

04 Tool call output and display controls in CLI IMPROVED

90

Control how much tool call information is shown by running `/settings output-details` and choosing the `collapsed`, `lean default`, or `full` preset, or set tool arguments, reasoning, output filters, subagent detail, and truncation individually. Tool calls now share a consistent look: shell, search, and generic output folds under the tool call corner, finished calls show returned file lines, two or more consecutive finished tool calls collapse into one summary row, file-write approvals show a full diff instead of a capped preview, and finished searches keep their result count in scrollback.

Tune tool call verbosity mid-session to see only the information you need without leaving the chat.

📍 `/settings` output-details

— Exact command, presets, and an itemised list of display changes.

CLI: Output Details and Sign-In Fixes (…

05 New kirocrew CLI management commands NEW

90

Run `kirocrew cron add update` with `--auto-pause-after` to set a failure-backoff limit from the command line; run `kirocrew spawn run` with `--model`, `--agent`, and `--crew` flags to control which model and agent a spawned run uses; run `kirocrew agent create` to create a new agent from the command line; and run `kirocrew kas remove` with `--version` to remove a specific installed Kiro Agent version (a blank `--version` now refuses instead of removing all versions).

Keep remote MCP sign-ins alive across sessions so you are not prompted to re-authenticate on every new Kiro Agent session.

```
$ kirocrew cron add update --auto-pause-after 3
```

Spawn a run targeting a specific model, agent, and crew from the command line to automate a recurring engineering task.

```
$ kirocrew spawn run --model claude-sonnet-4-5 --agent my-agent --crew my-crew
```

— Four exact, directly runnable subcommands and flags.

Crew: Kiro Agent by Default and Chat Yo…

06 **Mermaid diagram viewing improvements** IMPROVED 85

In IDE chat, Mermaid diagrams now stop shrinking at 80% of their natural size so labels stay readable, support drag-to-pan for larger diagrams, and show an easier-to-read zoom toolbar on hover. In the CLI, wide diagrams open in a full-screen viewer via `Ctrl+]` or the `/diagram` command, with arrow keys or mouse wheel to scroll, Tab to step through older diagrams, and Esc to return to chat; the viewer's keybinding can be changed with `chat.keybindings.viewDiagram`.

When an agent reply contains a Mermaid diagram too wide for your terminal, open it at full width to scroll and inspect it.

```
$ /diagram
```

— Exact keybindings and command given for the CLI viewer.

IDE: Follow Workflow Runs, Skills After… | CLI 3.2.1: Full-Screen Diagram Viewer a…

---

07 **Kiro Crew Bots: persistent, schedulable agents (formerly Mates)**

NEW 85

Kiro Crew adds bots — named, persistent agents each with their own memory, chat, notes, and workspace — so recurring responsibilities are owned by a dedicated agent instead of tracked across isolated sessions. The feature was first introduced as mates acting as conductors with their own worker sessions, enabled via Settings → Developer → Feature Previews → Mates, with each mate's agent file stored under `~/.kiro/mates/agents/` and importable from a `.crewmate` file for portability across installs. Each bot can run sessions in parallel across worker agents, monitor and redirect work that drifts from the plan, surface only findings and decisions needing attention to its ongoing chat, run on schedules set in its chat (e.g. overnight CI checks or release notes), hold independently scoped tool access and permissions (e.g. a research bot that can read but not write files, or a deployment bot requiring approval before each change), maintain its own notes and work log, show a customizable dashboard, and run on a model chosen separately from its worker agents' model. Start by opening the bots workspace in the Kiro Crew sidebar, adding a bot, and choosing the default or a custom agent as its base. In Crew 0.12, Mates were renamed to Bots across every screen, prompt, and command bar row; older `.crewjam` and `.crewmate` export files still import.

Create a build-health bot that checks your CI pipeline on a schedule and brings back only new failures that need attention.

📍 1. In Kiro Crew, open the bots workspace in the sidebar. 2. Select Add a bot and give it a name, e.g. Build Health. 3. Choose a custom agent as its base and

scope its tools to CI pipeline access only (no write permissions). 4. In the bot's chat, type: 'Check the CI pipeline every 30 minutes. Sort new failures from known flaky tests and only bring back what needs my attention.' 5. The bot confirms the schedule; scheduled activity goes to its work log, findings return to chat.

Split planning and routine work across two model tiers inside a single bot to keep costs down while preserving quality on complex decisions.

📍 1. Open the bots workspace in the Kiro Crew sidebar and select your bot. 2. In the bot's settings, choose a stronger model (e.g. a frontier model) as the bot's own model for planning. 3. Set a faster, cheaper model as the worker agent model for routine checks. 4. The bot now plans on the stronger model and delegates execution to the lighter one automatically.

Enable the conductor-and-worker mates preview so each mate runs hands-on work in a dedicated worker session.

📍 1. Open Settings › Developer › Feature Previews 2. Enable `Mates` 3. Existing mates are migrated to ~/.kiro/mates/agents/ automatically on next startup

— Rich mechanism and setup steps, though no API/CLI surface named. Exact settings path, file location, and import format.

launch-20261105-2f4f22ef | Crew: Bots for Everyone and a Browser Y… | Crew: Kiro Agent by Default and Chat Yo…

---

08 **Memory across chat sessions in IDE** NEW 80

Kiro adds a Memory view, accessible via the Memories icon in the activity bar, where you can review what the agent remembers across sessions — including repository conventions, corrections, and approaches already ruled out. Control how the agent reads and writes memories with the `Kiro Agent: Memory` setting, including a `Use only` mode that lets Kiro use memories without changing them; chat now shows when Kiro accesses or changes a memory during a session.

Prevent the agent from adding new memories while still benefiting from existing ones — useful when you want stable, curated context in a shared or sensitive project.

📍 In Kiro IDE, open Settings and set `Kiro Agent: Memory` to `Use only`. The agent will read existing memories but will not create or modify any during the session.

Prevent Kiro from writing new memories while still letting it apply existing ones — useful in shared or regulated environments.

📍 In the IDE, open Settings and set `Kiro Agent: Memory` to `Use only`.

— Names the exact setting and mode with a runnable configuration step.

IDE 1.4.178: Memory Across Chat Session… | IDE 1.4.121: Memory Across Chat Session…

---

09 **Expanded kiro-cli login options** NEW INTEGRATION IMPROVED

80

Sign in to an Identity Center instance from the CLI by running `kiro-cli login --license pro` with `--identity-provider` and `--region`. On Linux hosts with no graphical session, `kiro-cli login` at a terminal now uses the device flow and prints a code and URL instead of attempting a browser-based sign-in.

Authenticate to a specific AWS Identity Center instance from a headless CI or server environment.

```
$ kiro-cli login --license pro --identity-provider <provider-url>
--region <aws-region>
```

— Exact command and flags for both headless and Identity Center sign-in.

CLI: Output Details and Sign-In Fixes (…

---

10 **Notification settings in Crew** NEW

80

Turn on push notifications that fire after the app is closed via Settings → Notifications → Push notifications → Notify me even when Kiro Crew is closed, including iOS delivery after Add to Home Screen, and set a custom notification sound in any category via Settings → Notifications → Custom sounds, accepting short MP3, WAV, or OGG files.

Receive Kiro Crew approval alerts on your phone even after closing the browser or desktop app.

> 📍 In Kiro Crew, go to Settings -> Notifications -> Push notifications and enable 'Notify me even when Kiro Crew is closed'. On iOS, also tap Share -> Add to Home Screen first so the web app can receive alerts.

— Exact settings paths and accepted file formats named.

Crew: Bots for Everyone and a Browser Y…

---

11 **Browser automation controls and live-view takeover in Crew**

IMPROVED

80

Pick which browser the agent drives and control the live view, its quality, frame rate, and recording via Settings → Browser, replacing the former built-in browser switch and extension token field. When the agent uses the browser, the page streams live into chat so you can click the picture to take over and hand the browser back when done, with a secret-entry sheet so credentials are never seen by the agent.

Let the agent handle a login flow without exposing the password: watch the browser live and hand credentials over via the secret-entry sheet.

> 📍 In Kiro Crew, go to Settings -> Browser, choose your preferred browser (Chrome, Edge on Windows, or Kiro's own Chromium) and turn the live view on. When the agent reaches a sign-in page, the chat shows the page live; click the picture to take over or use the secret-entry sheet to type credentials — nothing typed is saved and the agent never sees it.

— Names the config path and the before/after of replaced controls.

`Crew: Bots for Everyone and a Browser Y…`

---

12 **Kiro Mobile for Android** NEW

75

Kiro Mobile is now available on Google Play, letting you start, follow, and steer Kiro cloud sessions from an Android phone without keeping a laptop running. It picks up sessions started in Kiro Web, the IDE, or the CLI with connected repositories and model choices already present; enabling `Build with spec` has the agent plan before building (in Autonomous mode the model selector shows `Auto` and cannot be changed); each edited file appears as a phone-sized diff card with the session's pull request status (e.g. merged or open), and you can attach images and answer agent questions inline while following live replies and tool activity.

Pick up an in-progress cloud session on your Android phone after starting it in the IDE or CLI, without leaving your laptop running.

> 📍 1. Install Kiro Mobile from Google Play. 2. Sign in with your Kiro account (Pro, Pro+, Pro Max, or Power plan required). 3. Open the session list — sessions started in Kiro Web, the IDE, or the CLI appear automatically with their repositories and model choices. 4. Tap a session to follow live replies and tool activity, attach images, or answer the agent's questions.

— Full feature list but only app install/sign-in steps, no API.

`Mobile: Kiro Mobile for Android (2026-1…`

---

13 **Amazon Bedrock Knowledge Base integration in Crew** NEW

75

Register an Amazon Bedrock Knowledge Base under `Capabilities → Knowledge → Sources → Add Knowledge Base` to query it live and merge results into knowledge search, with a cost-confirmation card shown before the first query.

Attach an Amazon Bedrock Knowledge Base so the agent can search your internal docs before the first query commits any cost.

📍 In Kiro Crew, go to Capabilities -> Knowledge -> Sources, click Add Knowledge Base, select your existing Amazon Bedrock Knowledge Base, confirm the AWS account and Region on the cost-confirmation card, then run a chat that references internal documentation.

— Exact UI path and cost-confirmation mechanism named.

Crew: Bots for Everyone and a Browser Y…

14 **Chat settings refinements in Crew** IMPROVED 75

Fine-tune the auto-compact threshold to any exact percent from 5 to 90 under Settings → Chat → Advanced, disable the jump-to-bottom behavior on send via Settings → Chat → Jump to the bottom when you send, and enable copy-on-select for chat transcripts via the opt-in Settings → Chat → Transcript → Copy on select.

— Three exact settings paths with concrete ranges and defaults.

Crew: Bots for Everyone and a Browser Y…

15 **KIRO_DEFAULT_SETTINGS environment variable** NEW 75

Set `KIRO_DEFAULT_SETTINGS` to the absolute path of a root-owned settings file on macOS or Linux to ship default configuration as the lowest-priority layer, overridden by each user's own `cli.json`.

Lock down a shared dev machine so every Kiro user inherits a common baseline config (e.g. approved model settings) without being able to accidentally override it at the system level.

```
$ export KIRO_DEFAULT_SETTINGS=/etc/kiro/defaults.json
```

— Exact env var and layering behaviour, with a runnable example.

CLI: Compaction Hooks and Session Credi…

16 **AWS CloudTrail audit logging for MCP tool calls** NEW 70

Kiro now records calls to Kiro service MCP tools as AWS CloudTrail management events by default for profiles in `us-east-1` and `eu-central-1`, capturing who made the request, which tool was called, when it ran, and whether it completed. Administrators can additionally log tool listings and read-only tool calls by enabling CloudTrail data events.

— Names regions and event types but no exact enablement steps.

General: Audit Kiro Tool Calls with AWS…

17 **Approve For Me commands in Command Palette** NEW 70

New 'Kiro: Approve For Me (User)...' and 'Kiro: Approve For Me (Workspace)...' commands in the Command Palette show the current state of Approve for me for new chats and let you toggle it on or off per scope (user or workspace).

Toggle Approve for me on or off per scope (user or workspace) without leaving the IDE.

> 📍 Open the Command Palette and run 'Kiro: Approve For Me (User)...' to see and change the setting for new chats at user scope, or 'Kiro: Approve For Me (Workspace)...' for workspace scope.

— Exact command names and toggle scopes are given.

IDE: Follow Workflow Runs, Skills After…

---

18 **New agent harness choices: Strands and GitHub Copilot CLI** NEW 70

Kiro Crew adds Strands and GitHub Copilot CLI as agent harness choices under Settings → Agent Harness (Strands requires the Strands CLI installed, with model provider keys stored in Settings → Secrets), and every harness's model picker now shows that backend's own model names.

— Named harnesses and settings path, with an install prerequisite.

Crew: Bots for Everyone and a Browser Y…

---

19 **New feature previews in Crew** NEW 70

Off-by-default preview switches under Settings → Developer → Feature Previews add Ultra Mode (the renamed, redesigned Crew Mode Board), Explore, Agent harness per Bot, Shared Crew, and Remote dispatch; Shared Crew lets the owner share their machine over Tailscale via a two-step card under Settings → Developer → Feature Previews → Shared Crew after a restart.

— Names all five previews and the Tailscale-sharing mechanism.

Crew: Bots for Everyone and a Browser Y…

---

20 **MCP sign-in via Client ID Metadata Documents** NEW 65

MCP sign-in now works without an `oauth` block when the authorization server supports Client ID Metadata Documents, offers no dynamic client registration, and accepts any port on a localhost callback — Kiro signs in using its own hosted client metadata document. This covers servers such as Tableau's hosted MCP, and applies in both the CLI and IDE.

— Named mechanism and example server, but no manual config steps.

IDE: Follow Workflow Runs, Skills After… CLI 3.2.1: Full-Screen Diagram Viewer a…

---

21 **Session management UI improvements in Crew** IMPROVED 65

Pop the current session into its own window with `Cmd+O` (or `Ctrl+O` on Windows and Linux), filter and group sessions with new `Group by kind` and `Not in progress` options in the sidebar Filter menu, and clean up sessions with new 14-day and 30-day windows.

— Exact shortcut and filter option names given. `Crew: Bots for Everyone and a Browser Y…`

---

22 **Settings > Developer reorganization in Crew** BREAKING

65

The toggle between Kiro Agent and Kiro CLI moved to `Settings → Developer → Feature Previews` in Crew 0.11, replacing its previous location. In 0.12, Developer Mode was removed entirely, so `Settings → Developer` now always shows Feature Previews, Logs, System, Telemetry, Storage, Memory, Config, Debug tools, and Archive.

— Exact before/after menu contents named across two releases.

`Crew: Bots for Everyone and a Browser Y…` `Crew: Kiro Agent by Default and Chat Yo…`

---

23 **Persistent remote MCP sign-in setting** IMPROVED

65

Set `agent.kas_persist_mcp_sign_in` to keep remote MCP sign-ins across sessions without re-authenticating each time — off by default in Crew 0.11.0, on by default from 0.11.2.

— Exact config key and default-value history across patch versions.

`Crew: Kiro Agent by Default and Chat Yo…`

---

24 **Per-job scheduling controls in Crew** NEW

65

Configure per-job schedule limits using the Schedule form: a `Time limit` (default 30 minutes) and an 'Auto-pause after N failures in a row' count, where 0 means never pause. Set `strict_schedule` on an app's cron to fire exactly on schedule with no spread.

— Named form fields, defaults, and a config key for exact timing.

`Crew: Kiro Agent by Default and Chat Yo…`

---

25 **Session recap in CLI** NEW

65

Enable session recap by turning on `Session recap` under `/settings > Features` or setting `chat.enableSessionRecap`, to see a short recap above the prompt after idle seconds.

Turn on the session recap so a summary appears above the prompt after you step away, without manually running `/usage`.

> 📍 In the Kiro CLI, run `/settings` › Features and enable `Session recap`.

— Named setting and config key with a clear UI path.

`CLI: Compaction Hooks and Session Credi…`

26 **Session credit tracking in CLI** NEW 65

Track session credit consumption in `/usage`, in the status line via the Session credits segment under `/settings display`, and per step in the workflow monitor.

— Three named surfaces where credit tracking now appears.

CLI: Compaction Hooks and Session Credi…

27 **Sub-agent visibility and memory access** IMPROVED 60

Each sub-agent's assigned task now shows in the inline sub-agent panel and in the Ctrl+G agent monitor, which also marks the sub-agent's final response. Sub-agents can also read the memories of the agent that dispatched them, read-only — they cannot add, change, or delete the parent's memories.

— Names the monitor shortcut; memory access is explicitly read-only scoped.

CLI: Copy Earlier Responses and Choose … CLI: Compaction Hooks and Session Credi…

28 **Steering document warnings in Agent Steering view** NEW 60

The Agent Steering view in the Kiro IDE sidebar now flags steering documents it could not fully apply with a warning icon and problem count; hovering shows each problem description and a suggested fix so you can edit the document to resolve it.

Identify and fix steering documents that Kiro could not fully apply before they silently affect agent behaviour.

📍 1. Open the Agent Steering view in the Kiro IDE sidebar. 2. Look for any steering document marked with a warning icon and a problem count. 3. Hover over the warning to read each problem description and the suggested fix. 4. Edit the steering document to resolve the flagged issues.

— Clear UI navigation path but no named config surface.

IDE: Workflow Run Tab and Steering Warn…

29 **Bot export and import as .kirobot files** NEW 60

Export a Bot's profile as a `.kirobot` file via Export as file on its profile card, and re-import it via Bots → Import from file.

— Exact file format and menu path, minimal further detail.

Crew: Bots for Everyone and a Browser Y…

THINNER COVERAGE BELOW

30 **Skill and steering file reload after compaction** IMPROVED 55

After Kiro compacts a conversation, the agent now reloads the skills and steering files the summary dropped, loads a large skill or steering file in full when it fits instead of as a

cut-off preview, and skips re-adding a skill already present in the conversation.

– Explains the mechanism but has no user-facing command.

IDE: Follow Workflow Runs, Skills After…

31 **Kiro Agent now default harness in Crew** BREAKING 50

Kiro Agent is now the default harness; a new install with no saved harness choice runs new sessions on Kiro Agent, while a previously chosen harness (including Kiro CLI) is preserved. The desktop app no longer bundles Kiro CLI — it ships the Kiro Agent harness it was tested with and detects a Kiro CLI you installed yourself.

– Explains the default change and bundling impact, no migration steps.

Crew: Kiro Agent by Default and Chat Yo…

32 **Bitbucket Server hosts for PR sources** NEW 50

Add Bitbucket Server hosts for pull-request sources via the `dashboard.bitbucket_server_hosts` setting.

– Names the config key but no further mechanism described.

Crew: Kiro Agent by Default and Chat Yo…

33 **Markdown-preferring web fetch in IDE and CLI** IMPROVED 50

Web pages Kiro fetches now request Markdown or HTML so that sites offering Markdown can preserve code blocks and links; in the CLI this applies to `web_fetch` requests to documentation pages.

– Names the affected tool but no configuration flag given.

IDE: Compaction Hooks, Workflow Run Sea… CLI: Compaction Hooks and Session Credi…

34 **Color-blind safe and Amethyst OLED themes** NEW 45

Apply Color-blind safe or Amethyst OLED themes via Settings → Display → Theme.

– Named themes and setting path, nothing further described.

Crew: Bots for Everyone and a Browser Y…

35 **file:// resource caps in CLI** BREAKING 45

Agent `file://` resources are now capped at 32 files and 100,000 characters, with large files cut or skipped.

– Exact numeric limits given, but no mitigation steps.

CLI: Compaction Hooks and Session Credi…

36 **MCP Management relocated to Connections page** BREAKING

40

MCP Management moved from its previous location to the Connections page, below the MCP Servers list.

— States the new location only, no further detail.

Crew: Bots for Everyone and a Browser Y…

---

37 **Summon hotkey off by default on new installs** BREAKING

40

The summon hotkey is off by default on new installs; turn it on in Settings. Existing installs keep their previous setting.

— Short note with a settings pointer but no exact path.

Crew: Kiro Agent by Default and Chat Yo…

---

38 **MCP credential file permission hardening** IMPROVED 40

On macOS and Linux, remote MCP registration and credential files are now readable and writable only by the owning user, and new credential folders are created with owner-only permissions.

— Describes the security change; no user action required or given.

CLI: Compaction Hooks and Session Credi…

---

39 **Usage tab shows provider credits instead of estimated tokens**

35

BREAKING

The Usage tab no longer shows estimated tokens — the Estimated Tokens card and 'Est. tokens' column are removed — and now shows credits reported by the provider.

— Describes removal and replacement, no further mechanism.

Crew: Kiro Agent by Default and Chat Yo…

---

40 **Notification queue consolidation in CLI** IMPROVED 35

Notifications now queue into a single row, most urgent first, with a `+N` waiting count; warnings and errors stay on screen until dismissed.

— Describes new UI behaviour only, no configuration surface.

CLI: Compaction Hooks and Session Credi…

---

41 **About dialog shows Kiro Agent version** IMPROVED 25

The About dialog now lists the Kiro Agent version below the build date.

— Single-line cosmetic addition with no further detail.

IDE: Workflow Run Tab and Steering Warn…

└──▷ **BREAKING ON UPGRADE**

! Mates are renamed to Bots across every screen, prompt, and command bar row; older `.crewjam` and `.crewmate` export files still import.

Our best guess

Check this claim Claim history

Crew: Bots for Everyone and a Browser You Can Take Over (2026-11-05)

! Developer Mode is removed; `Settings → Developer` now always shows Feature Previews, Logs, System, Telemetry, Storage, Memory, Config, Debug tools, and Archive.

Our best guess

Check this claim Claim history

Crew: Bots for Everyone and a Browser You Can Take Over (2026-11-05)

! MCP Management moved from its previous location to the Connections page, below the MCP Servers list.

Our best guess

Check this claim Claim history

Crew: Bots for Everyone and a Browser You Can Take Over (2026-11-05)

! `Settings → Browser` replaces the former 'Use the built-in browser' switch and the extension token field.

Our best guess

Check this claim Claim history

Crew: Bots for Everyone and a Browser You Can Take Over (2026-11-05)

! Crew Mode is renamed Ultra Mode and remains an off-by-default Feature Preview with a redesigned Board.

Our best guess

Check this claim Claim history

Crew: Bots for Everyone and a Browser You Can Take Over (2026-11-05)

! Kiro Agent is now the default harness; a new install with no saved harness choice runs new sessions on Kiro Agent. A previously chosen harness, including Kiro CLI, is preserved.

Our best guess

Check this claim Claim history

Crew: Kiro Agent by Default and Chat You Can Stop (2026-10-30)

! The desktop app no longer bundles Kiro CLI; it ships the Kiro Agent harness it was tested with and detects a Kiro CLI you installed yourself.

Our best guess

Check this claim Claim history

Crew: Kiro Agent by Default and Chat You Can Stop (2026-10-30)

! The toggle between Kiro Agent and Kiro CLI now lives in Settings > Developer > Feature Previews instead of its previous location.

Our best guess

Check this claim Claim history

Crew: Kiro Agent by Default and Chat You Can Stop (2026-10-30)

! The summon hotkey is off on new installs; turn it on in Settings. Existing installs keep their setting.

Our best guess

Check this claim Claim history

Crew: Kiro Agent by Default and Chat You Can Stop (2026-10-30)

! The Usage tab no longer shows estimated tokens; the 'Estimated Tokens' card and 'Est. tokens' column are removed. The tab now shows credits reported by the provider.

Our best guess

Check this claim Claim history

Crew: Kiro Agent by Default and Chat You Can Stop (2026-10-30)

> General: Credit pooling now works with Okta and Microsoft Entra ID (2026-11-10)
>
> Kiro adds Okta and Microsoft Entra ID support for enterprise credit pooling, with a Switch to credit sharing wizard and default pooling for new IdP-connected profiles.
>
> NEW INTEGRATION Share Kiro credits across a team when using Okta or Microsoft Entra ID as an identity provider, via Access management > Credit sharing in the Kiro console — in addition to the existing AWS IAM Identity Center support.
>
> Move existing individual subscribers to a shared pool using the Switch to credit sharing wizard in the Kiro console.
>
> New profiles connected to IAM Identity Center, Okta, or Microsoft Entra ID now start with credit pooling enabled by default.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

05 fizgig v9.2.1

Model Training & Fine-Tuning

Fizgig — LoRA & Fine-tune Studio for Klein 9B, Krea 2, MiniMax H3, Qwen Image 2.1, Anima & Z-Image Turbo and Ace-Step XL: train, fine-tune, profile, repair and extract LoRAs & LoKR for Image, Video & Music Models

// WHY IT MADE THE LIST — DEPTH 5/5 · IMPACT 5/5 · NOVELTY 4/5

PyTorch 2.15 integration cuts MOSS-Image training VRAM from 21.7 GB to 7.1 GB at batch 3 and makes Muon optimizer steps 12× faster with identical results, while the same release adds MiniMax H3 Image and Vimoo 1.0 Lite video model training — together making a class of fine-tuning workloads newly practical on consumer hardware.

> Fizgig added MiniMax H3 Image as a full training target and iterated on it across five releases, upgraded to PyTorch 2.15 for major training speed and memory reductions, switched likeness scoring to AdaFace ViT, and made Muon the default optimizer across eleven image models, alongside a run of training-workflow, queue-resilience and MOSS-Image preview improvements.

└──▷ WHAT SHIPPED · 22 FEATURES most completely described first ? what's the number?

01 MiniMax H3 Image training support NEW — 100

Fizgig added MiniMax H3 Image as a training target supporting LoRAs, LoKRs, edit LoRAs from before/after pairs, and slider LoRAs from photo pairs, with masked training, Multi Concept, DOP, Fizgig LoRA Isolation, batch size and Compile Blocks; a built-in training adapter is on by default, downloads with the model, stays frozen during training and is not saved into the output LoRA. A Clean-end share setting in Training Parameters controls the fraction of training spent on final denoising steps, defaulting to 40% versus the model's own ~17% default. Presets moved from 0.5 MP to 0.25 MP at 30 epochs; Element boost (from MOSS-Image) lets you weight face, hair or another part more heavily early on, off by default, enabled via the On tick in Training Parameters; Studio and Samples now default to 2048×2048 previews for H3 Image. A Likeness optimization shifts training weight toward the final steps where skin, eyes and fine detail resolve, on by default in the MiniMax H3 Image Standard preset, replacing the Clean-end share box (disable under Training Parameters to restore even share). Automagic v3 starts at `2e-4` skipping the initial climb phase, then Automagic v4 (starting at `2.5e-4`, capped via `max_lr=` in Optimizer Args) became the Standard preset's optimizer, beating Automagic v3 and Muon in likeness tests especially on datasets with few headshots, paired with a new likeness optimization schedule under Advanced inside the Likeness optimization box. Training adapters moved to Ostris's v3, with the updater migrating Preferences from v1 or v2 and leaving the old file on disk.

Tighten Automagic v4's per-weight learning rate ceiling below the default 2.5e-4 when you want more conservative updates on a sensitive dataset.

```env
max_lr=1.5e-4
```

— Exact settings, toggles and config key named across five releases v9.6.0 v9.6.1 v9.6.4 v9.7.0 v9.7.1

02 **MOSS-Image Turbo Train preset and Clean-end share scheduling**

NEW

85

A new MOSS-Image Turbo Train (Experimental) preset uses rank 16, 0.25 MP, Muon, Adaptive LR between `2e-4` and `4e-4`, per-image tracking, and per-image adaptive LR so each image gets its own learning-rate pace during training. MOSS-Image's Clean-end share order is now `30:7:60:8:3` for LoRA, Style, and Slider presets — 30% of steps on nearly clean images until epoch 7, then 60% from epoch 8, with every 3rd epoch after that at 30% and half the learning rate; Fine-tune keeps `11:4:30:8:2`.

— Exact preset parameters and schedule ratios named v9.2.5

03 **Raised system requirements for PyTorch 2.15** BREAKING

85

Python 3.10 is no longer supported since PyTorch 2.15 has no Python 3.10 build, so installs on Python 3.10 cannot update — upgrade to Python 3.11, 3.12 or 3.13 and run `install_fizgig.bat` (Windows) or `python3.12 install_fizgig.py` (Linux) to rebuild the environment. An NVIDIA driver that supports CUDA 13 (version 580 or newer) is now required; if Fizgig cannot see your GPU after updating, update the NVIDIA driver. RunPod pod images move to Ubuntu 24.04 (Python 3.12) to carry PyTorch 2.15, and existing pods must be restarted to pick up v9.2.

— Exact version requirements and migration commands given v9.2.1

04 **ROCm nightly channel and AMD stable default** IMPROVED

80

Use `ROCM_CHANNEL=nightly` on Linux or `install_fizgig_rocm.bat --experimental` on Windows to keep nightly ROCm builds; AMD installs now default to torch 2.14.0 for ROCm 10.1 from AMD's stable index.

Keep ROCm nightly builds on a Linux AMD system instead of adopting the new stable default.

```
$ ROCM_CHANNEL=nightly ./install_fizgig.sh
```

Keep ROCm nightly builds on a Windows AMD system instead of adopting the new stable default.

```
$ install_fizgig_rocm.bat --experimental
```

— Exact commands and version numbers given for both platforms v9.6.0

05 **PyTorch 2.15 training speed and memory improvements** IMPROVED

80

Training now uses far less GPU memory with gradient checkpointing on PyTorch 2.15: Krea 2 drops from 22.9 GB to 14.9 GB, MOSS-Image at batch 3 from 21.7 GB to 7.1 GB. The Muon optimizer now processes a LoRA's matrices together instead of one at a time, making each step 12× faster with identical results. Batched steps are faster when captions vary in length: 18% faster on MOSS-Image, 12% on Krea 2, 9% on Qwen Image 2.1, and 8% on Z-Image. LoKR training is now 4–6× faster on every model, including LoKR Context LoRA runs and LoKR files in Studio and other tools. Studio's Show early feature on MiniMax H3 now uses `taeh3`, a 22.7 MB tiny H3 decoder, for mid-render previews roughly 10× faster than the full 4.5 GB decoder. Switching a preset from Muon or Prodigy to AdamW or AdamW8bit now clears Optimizer Args automatically, preventing silent fallback to slower plain AdamW.

— Many concrete before/after numbers but mostly automatic v9.2.1

06 **AdaFace-based likeness and style scoring** IMPROVED 78

Likeness scoring switched from ArcFace to AdaFace ViT across every tool, and existing ArcFace scores are rescored the first time you open them (reference photos must be picked again per training folder). Fizgig now fetches only InsightFace's face finder (18 MB) plus AdaFace ViT (460 MB) instead of InsightFace's full 300 MB pack. Likeness and Style scores are now stored per run and per training folder so reference photos are chosen once per folder and reused across every run without mixing or overwriting between runs sharing a samples folder. The Samples gallery now scores only the run currently being viewed, scoring new previews immediately, and its likeness chart adds a green your photos band showing the middle half of training photos' likeness scores with a dashed median line and a plain-language summary.

.

— Names the scoring engine, file sizes and storage scope; triggered by opening a run

v9.7.0 v9.7.1

07 **Muon optimizer default across image models**

NEW SUPPORTED PLATFORM IMPROVED 75

Fizgig now defaults every LoRA, slider, and edit preset to the Muon optimizer for Klein 9B, Flux 3 Image Klein, Fibo 2.0, Krea 2, Qwen Image 2.1, MOSS-Image, Z-Image Turbo and Base, SDXL, Anima, and Vimoo 1.0 Lite — reload any preset to apply it; previously

saved presets and the Training form keep the optimizer they had. Fibo 2.0's main preset now trains at a flat learning rate of `1e-4` instead of Automagic's `1e-6` start, though Automagic remains selectable in the Optimizer Type row. Muon uses less memory than AdamW and slightly more than AdamW8bit (about 0.2 GB on a rank-32 Krea 2 LoRA) by storing one value per trained parameter instead of two; AdamW8bit and AdamW remain available in the Optimizer Type row.

— Lists every affected model and exact memory/LR figures v9.0.8

08 **Per-job tab switching in training UI** NEW 70

Each job's training folder, captions, masks, sample prompts, settings and previews now live in their own view, switchable from a pill in the top right (bottom right on phone), so you can move between the active run, queued runs and a new setup without losing place. You can also set up a new run from the pill's list on the Training tab to copy the current job into a new setup with the same training folder, named with `_2` (or the next free number) after the original name.

— Describes mechanism and exact UI location but no command v9.7.1

09 **MOSS-Image preview speed-ups** NEW 70

A Faster, smaller previews (768 px) tick for MOSS-Image cards under 20 GB was introduced off by default (the gallery marks previews made at the smaller size), then switched to on by default with an option to untick for full-size previews. Separately, the 'Use the MOSS-Image Turbo LoRA for previews' tick on the Samples tab, with a Strength control beside it, swaps in a faster Turbo LoRA for preview generation, followed by the same setting in Studio, LoRA Royale and the Profiler.

— Names both UI controls and their default states exactly v9.2.2 v9.2.5 v9.2.4

10 **CUDA Sysmem Fallback startup diagnostic** NEW 70

Fizgig now shows startup steps on Windows with an NVIDIA card to set `CUDA - Sysmem Fallback Policy` to Prefer No Sysmem Fallback in NVIDIA Control Panel's Global Settings to restore full GPU training speed, with a Don't show this again option to dismiss it once configured.

— Exact setting name and navigation path to fix it v9.6.3

11 **Interrupted-run recovery on restart** NEW 65

Fizgig now detects a run cut off mid-training and, on the next start, asks whether to carry on from the last saved state, start the next queued run, or clear the queue and start fresh — before anything else runs. A line above Start appears when the Resume Training box holds a saved state, naming which state Start would continue from, with a button to clear it.

— Names the exact UI element and behaviour on restart v9.7.1

12 **Multi-line edit instructions with per-line preview** IMPROVED

60

The Edit instruction box on the Edit pairs card now accepts one instruction per line, and every line is previewed on your test photo each round, so a LoRA trained on a family of edits shows all of them at once.

— Explains mechanism and benefit, UI path only v9.2.4

13 **Out-of-memory auto-retry and queue resilience** NEW

60

Out-of-memory recovery now automatically retries a failed web-UI run up to three times, stepping down through more block swap, then lower precision, continuing from the newest save when one exists. A failed run no longer stalls the queue; Fizgig names the failed run in the Queue box and starts the next one automatically. Switching models on the Training tab now loads that model's first preset, clearing settings that do not apply to the new model while keeping folders and the Samples tab.

— Mechanism and fallback steps described, fully automatic v9.6.0

THINNER COVERAGE BELOW

14 **Starred preview checkpoints** NEW 50

Star any preview in the Samples gallery, the grid, or a run's card to mark its epoch's checkpoint with a star in the checkpoint lists, making it easy to find the right LoRA later.

— Clear UI action, limited mechanism detail v9.7.1

15 **Duplicate output-name guard** NEW 50

Start, Add to queue, and Check settings now refuse a LoRA name that already has files in the same output folder and tell you which name is free, preventing two runs from overwriting each other's files.

— Names the three guarded actions, mechanism is simple v9.7.1

16 **Draw window without graphics card preference** NEW 50

Add a 'Draw the window without the graphics card' option in Preferences > App window so the window holds none of the VRAM a tight run needs.

— Exact preference path given, behaviour explained briefly v9.7.1

17 **Run card status additions** IMPROVED 45

The run card now shows steps remaining in the current epoch beside the step counter, reports training speed excluding everything between epochs (saves, checks, and previews) so it reflects training-step speed only, and the run summary card shows the real batch size used during a run.

— Three named display additions to the same run card, display-only v9.2.1 v9.2.4 v9.7.1

18 **Pod compile cache persistence** IMPROVED 45

Pods now keep compiled blocks on the pod's volume so a Resume after a restart, Pause or Move reuses them instead of recompiling from scratch.

— Clear mechanism, but automatic with no user action v9.2.2

19 **Download progress for model files** IMPROVED 35

Fizgig now shows download progress for model files with speed and time remaining beside the size, and logs each filename as it starts.

— Describes the UI addition but no further detail v9.7.1

20 **Telemetry opt-out** IMPROVED 35

Fizgig now disables onnxruntime telemetry (device ID and machine details to Microsoft) and Hugging Face usage reporting on every platform.

— Names the two telemetry sources disabled, no user action needed v9.2.1

21 **Per-image loss view refresh** IMPROVED 30

Per-image loss view adds a Refresh button beside its sort menus, and changing either menu now loads the latest results automatically.

— Minor UI addition with little mechanism v9.2.5

22 **Caption box auto-sizing in Captions tab** IMPROVED 25

Caption boxes in the Captions tab now grow to fit their text in cards, the table, and the big view; on a wide window the big view puts the caption beside the picture.

— Thin cosmetic description with no mechanism beyond layout v9.7.1

└──▷ **BREAKING ON UPGRADE**

! Python 3.10 is no longer supported: PyTorch 2.15 has no Python 3.10 build, so installs on Python 3.10 cannot update. Upgrade to Python 3.11, 3.12 or 3.13 and run `install_fizgig.bat` (Windows) or `python3.12 install_fizgig.py` (Linux) to rebuild the environment.

Our best guess

Check this claim Claim history

v9.2.1

! An NVIDIA driver that supports CUDA 13 (version 580 or newer) is now required. If Fizgig cannot see your GPU after updating, update the NVIDIA driver.

Our best guess

Check this claim Claim history

v9.2.1

! RunPod pod images move to Ubuntu 24.04 (Python 3.12) to carry PyTorch 2.15; existing pods must be restarted to pick up v9.2.

Our best guess

Check this claim Claim history

v9.2.1

! Likeness and Style scores recorded before v9.7.1 are rescored the first time you open a run, and you must pick the three reference photos once more per training folder.

Our best guess

Check this claim Claim history

v9.7.1

v9.0.7

Fizgig adds Vimoo 1.0 Lite training, batch size Auto on 40 GB+ cards for five models, and a Studio deferred-render Update button.

└──▷ GET THIS VERSION ▾

Docker

```
$ docker pull ghcr.io/shootthesound/fizgig:9.0.7
```

git

```
$ git clone --branch v9.0.7
https://github.com/shootthesound/Fizgig.git
# already have the repo? check out this version:
$ git checkout v9.0.7
```

published under the project's GitHub account · checked 7 Nov

Batch size now has an Auto option for SDXL, MiniMax H3, Anima, Flux 3 Image Klein and Fibo 2.0 on 40 GB+ cards, picking the measured fastest batch per step and reporting the choice under the batch size box.

Gradient checkpointing now turns off automatically on 40 GB+ cards for SDXL, Anima, Flux 3 Image Klein and Klein when the run fits, running 1.3–2.3× faster.

Studio now holds renders until you click Update, letting you change size, clip length, steps, seed, reference picture, negative prompt and LoRA strengths together before triggering a render.

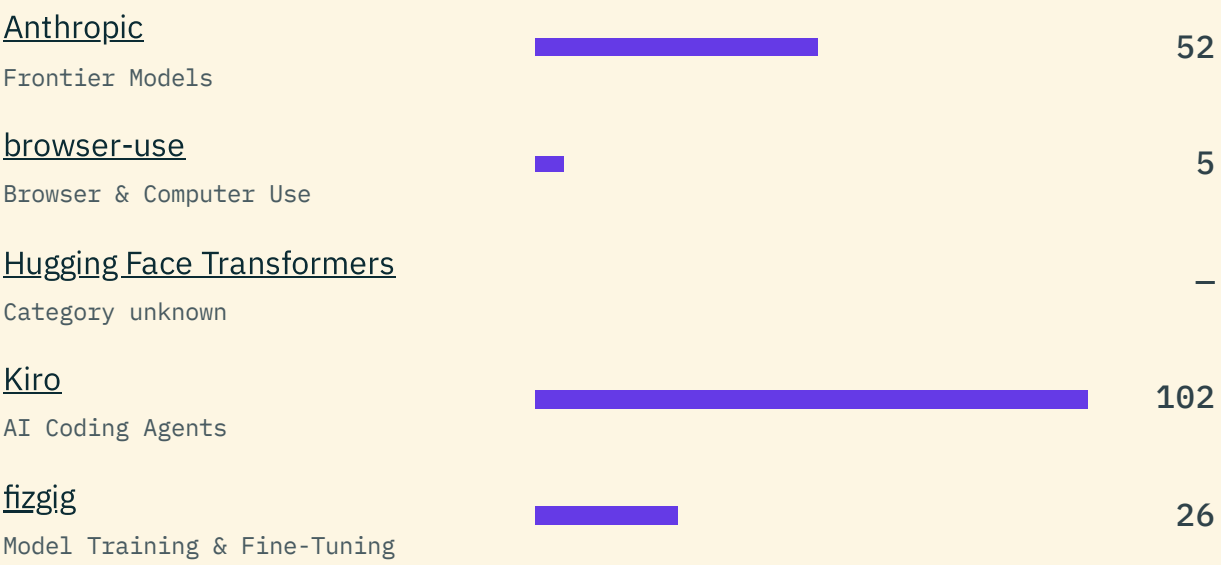
NEW SUPPORTED PLATFORM Gizmo now asks which video model a clip is for and prepares it to that model's frame counts and sizes.

> The Turbo Preview option in Studio is now called Reuse unchanged blocks and moves its stored results to system RAM when the card is tight, so long clips no longer run out of memory with it enabled.

GOOD PICK? 👍 👎 GOOD REASON? 👍 👎

Where each pick stands · Nov 7–13

Tools with changes in each pick's category over the last 12 weeks, including the pick itself.



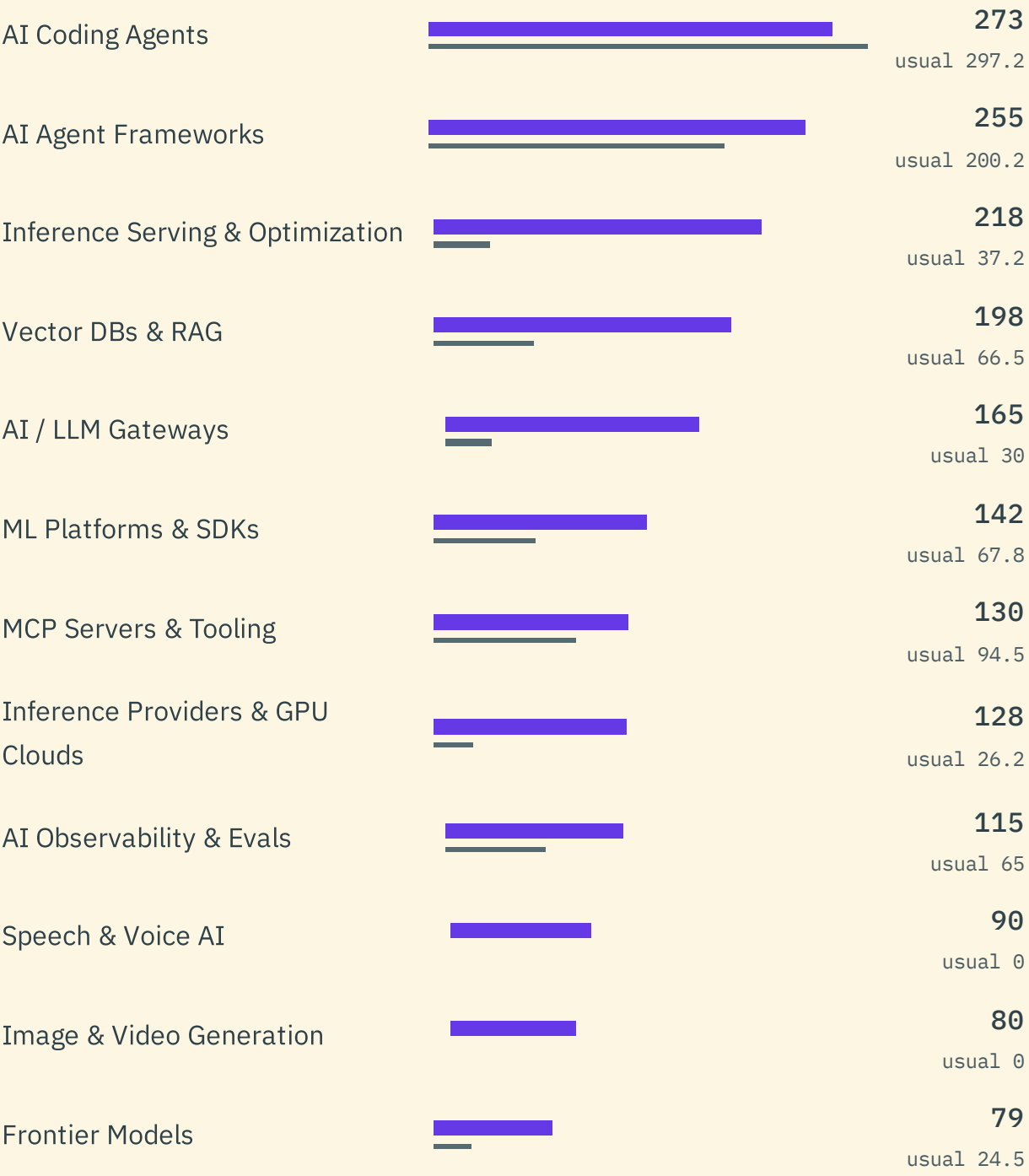
Each pick's release rhythm · Nov 7–13

Changes per week for the last 12 weeks. The bright bar is this week; a dash marks a week with no daily.



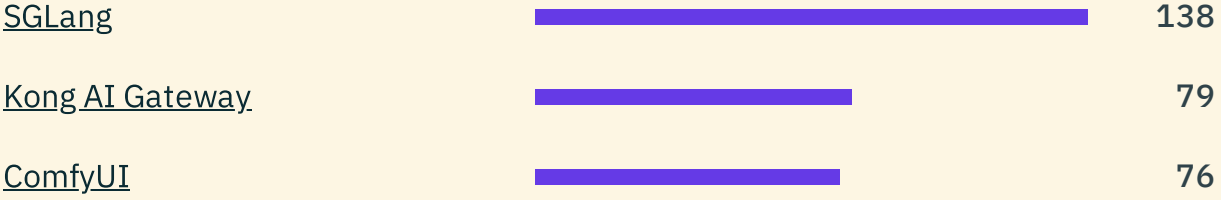
Which categories moved · Nov 7–13

The thick bar is this week. The thin bar is the average of the four weeks before, counting only weeks with a daily.



Who shipped the most · Nov 7–13

Changes per tool this week. Volume is not a quality score.



Pipecat		76
LiteLLM		65
magpie		64
Ray		48
Hermes		43

▾ // CONFERENCE CHANGES HIDE

Calls for papers, tickets, dates, talks and recordings this week. Star a conference to hear about the ones you follow.

The talks for PyTorch Conference 2026 are out.

Upcoming conferences →

// END OF TRANSMISSION

The AI Toolchain · This Week's Highlights · November 14, 2026
The full firehose — every feature release, every day — is in the newsletter.

▷ Subscribe