THE AI TOOLCHAIN

NO. 054

Tail

THE DAILY RELEASE FIREHOSE

PUBLISHED OCTOBER 10, 2026 · EVERY DAY

The daily firehose — everything the toolchain shipped today, already filtered.

▸ // HOW THIS ISSUE IS MADE SHOW

VIEW

ISSUE VIEW

scroll



SCROLL | EXPAND IN PLACE | PREVIEW PANE

INFORMATION DENSITY

everything



BRIEFED | EVERYTHING

Every tool in the issue, in full.

all 217 changes

Do you prefer **scroll**? 👍 👎

▸ // CONTENTS SHOW 53 tools

▾ // THE BRIEF HIDE

What stands out across today's releases, grouped by what it lets you do. Every tool named links to its entry below.

Three unrelated things stand out. FlashInfer's DeepSeek-V4 sparse MLA work adds NVFP4 KV-cache quantization and a dedicated dsv4 decode ...l, cutting what long contexts cost in HBM; LMCache now runs as a

vim ? help

standalone KV-cache server for SGLang over ZMQ or gRPC; and OpenViking's Gateway injects memory into any OpenAI- or Anthropic-compatible client via a base-URL swap.

DEPLOY

Serve long-context DeepSeek-V4 on fewer GPUs, and share a KV cache across SGLang replicas

FlashInfer's NVFP4 KV-cache quantization and dsv4 decode kernel shrink the per-token memory that normally caps concurrency on long prompts, while LMCache's SGLang integration moves the cache out of the engine process so a prefix computed on one replica can be reused by another instead of recomputed. SGLang also exposes EAGLE speculative-decoding block verification, which drafts tokens with a small model and verifies them in one pass on the big one.

FlashInfer · LMCache · SGLang

BUILD

Give an existing chat client persistent memory without touching its code

OpenViking's Gateway sits in front of any OpenAI- or Anthropic-compatible endpoint, so adding recall is a base-URL change rather than a rewrite of the call path. PydanticAI covers the same ground inside the framework with Conversation persistence on hosted Postgres plus cross-provider prompt caching controls, so a long standing system prompt stops being re-billed on every turn.

OpenViking · PydanticAI

GOVERN

Hand out scoped API keys and prove to an auditor what a pipeline did

fal's seven key presets with custom policies, model-scoped limits and expiry mean a key for one model no longer carries account-wide reach, and leaked keys age out on their own. Pipelock's Audit Packet format with a browser-based verifier lets someone outside your infrastructure check a recorder chain and `evidence.jsonl` without running your toolchain.

fal

BUILD

◆ AI Coding Agents

Cotool

Feedback ⓘ changelog · Oct 10, 2026 CHANGELOG ↗

Cotool is an AI security operations platform for investigating alerts, enriching threat intelligence, and automating response workflows.

Sub-agent delegation, suggested alert dispositions, per-agent sandbox network overrides, JSON schema editing, and a redesigned Settings UI land across v0.72 and v0.73.

Every agent and chat can now delegate focused subtasks to sub-agents.

Suggested dispositions on escalated alerts let analysts confirm environment statements from the alert page or Slack; later triage runs reuse these as context.

Per-agent sandbox network overrides and a clearer allowlist editor give fine-grained egress control per agent.

JSON editing for Agent Output schemas adds pasted-schema normalization and live Builder and Preview synchronization.

Controls to disable or delete output destinations remove the need to manually unlink them from every agent and notification.

+4 more

NEW INTEGRATION Proofpoint integration

NEW INTEGRATION proxycheck.io integration

UPDATED INTEGRATION Databricks integration rebuild

NEW INTEGRATION incident.io incident creation

NEW INTEGRATION HackerOne report management

UPDATED INTEGRATION Wiz integration expansion

WAS THIS USEFUL? 👍 👎

Augment Code

Feedback ⓘ snapshot-20261010 NOTES ↗ RANK

Augment Code is an AI coding assistant for VSCode and JetBrains IDEs that provides code review and context-aware suggestions, with Cosmos as its enterprise platform.

> Augment Code's Cosmos platform adds Azure DevOps integration with automatic organization discovery and webhook-based agent access, service accounts with API tokens for Enterprise and team plans, a GitHub Enterprise Server connector, and a Fleet Management bulk model update, alongside Linear connector improvements and a breaking change removing agent-initiated PR merges.

└──▷ **WHAT SHIPPED · 9 FEATURES** most completely described first ? what's the number?

01 Azure DevOps integration NEW

90

Cosmos can now connect to Azure DevOps via a bot account using a personal access token scoped to one organization, or via OAuth so Cosmos discovers every reachable Azure DevOps organization; once connected, agents gain repository access and respond to pushes and pull request activity through auto-configured webhooks.

Connect Azure DevOps to Cosmos via OAuth so agents can automatically discover and act on all reachable organizations without per-org token management.

> 📍 In the Cosmos console, go to Connectors › Azure DevOps, choose Connect with OAuth, authenticate, and Cosmos will enumerate every Azure DevOps organization the account can reach and configure webhooks for push and pull request activity.

— Mechanism, scope and setup steps all specified snapshot-20261010

02 Service accounts with API tokens NEW

70

On Enterprise and team plans, admins can create service accounts, issue API tokens for them, and inspect per-account access.

Create a service account and issue an API token so automated pipelines can call Cosmos without tying access to a personal user account.

> 📍 In the Cosmos console, go to Settings › Service Accounts, click Create service account, then select the new account and choose Issue API token; copy the token for use in your pipeline.

— Clear admin steps but no API endpoint named snapshot-20261010

03 Bulk model update in Fleet Management NEW

70

The Fleet Management page now supports a bulk model update that lets admins update the model for many Experts at once, using either an explicitly chosen model or each Expert's recommended model.

Bulk-update the model for a large Expert fleet in one operation rather than editing each Expert individually.

📍 In the Cosmos console, go to Fleet Management, select the Experts to update, click Update model, then choose either a specific model or Use each Expert's recommended model.

— Names feature and UI path with runnable example `snapshot-20261010`

THINNER COVERAGE BELOW

04 **Linear connector setup and reconnect prompts** IMPROVED 35

The Linear connector now walks admins through setting up Ticket Manager, and Cosmos prompts admins to reconnect Linear when its permissions are out of date.

— Describes behavior but no named config or flags `snapshot-20261010`

05 **GitHub Enterprise Server connector** NEW 30

Adds GitHub Enterprise Server to the Connectors catalog, including a setup guide.

— Thin description, no mechanism given `snapshot-20261010`

06 **Read-only IdP-synced groups and members** IMPROVED 25

Groups and members synced from an identity provider are now read-only in Cosmos.

— States change but omits affected scope or reason `snapshot-20261010`

07 **Double-click to copy inline code** NEW 25

Double-clicking inline code copies it to the clipboard.

— Simple UI action, fully described `snapshot-20261010`

08 **Session sidebar integration status indicator** NEW 20

The session sidebar now surfaces when a session owner's connection to an integration needs attention.

— Minimal detail, no mechanism or scope `snapshot-20261010`

09 **Agents no longer merge pull requests** BREAKING 15

Agents no longer merge pull requests.

— No replacement workflow or rationale given `snapshot-20261010`

└──▷ **BREAKING ON UPGRADE**

! Agents no longer merge pull requests.

WAS THIS USEFUL? 👍 👎

Vercel v0

Feedback ⓘ snapshot-20261010 NOTES ↗ RANK

The v0 generative UI platform builds web applications from natural-language prompts.

> v0 introduces a dedicated Code tab for browsing, editing, and annotating project files directly within the platform.

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

01 **Code tab with file management and editing** NEW 55

A new Code tab lets users explore projects via a file tree viewer, edit code, manage files and folders, and add comments as context for the next AI prompt.

Browse your project's file tree, make inline edits, and annotate code with comments that steer your next AI prompt — all without leaving v0.

📍 In v0, open your project and select the `Code` tab to view the file tree, edit files, manage folders, and add inline comments as prompt context.

— Describes what it does and UI path, no technical mechanism. snapshot-20261010

WAS THIS USEFUL? 👍 👎

yetone magpie

Feedback ⓘ v0.1.1157 CODE ↗ RANK

A macOS menu-bar app, TUI, and CLI that lists every AI coding agent on the machine and switches each one's model, such as Codex on DeepSeek or Claude Code on Kimi, by rewriting only that key in the agent's own config.

> The biggest window for magpie adds opt-in local gateway conversation recording with transcript retrieval, hardens the gateway against DNS rebinding and tightens remote-key requirements, and widens agent coverage with Snow CLI and Zed MCP integrations alongside bulk ChatGPT account import, per-category PII redaction switches, model reply-limit visibility, and a loop guard against runaway model replies.

└──▷ **WHAT SHIPPED · 34 FEATURES** most completely described first ? what's the number?

01 MCP server delivery to Zed NEW

95

Adds MCP server delivery to Zed, writing entries into the `context_servers` key of Zed's `settings.json` (JSONC), supporting both `command` / `args` / `env` (stdio) and `url` / `headers` (streamable HTTP) transports, while preserving existing user settings, comments, and extension-installed servers.

Connect an MCP server to Zed by having magpie write it into `context_servers` in Zed's `settings.json`, so Zed picks it up alongside your existing theme and buffer settings.

json

```
// ~/.config/zed/settings.json (managed by magpie)
{
  "theme": "One Dark",
  "context_servers": {
    "my-mcp": {
      "command": "npx",
      "args": ["my-mcp-server"],
      "env": { "API_KEY": "<value>" }
    },
    "remote-mcp": {
      "url": "https://mcp.example.com/mcp",
      "headers": { "Authorization": "Bearer <token>" }
    }
  }
}
```

— Exact config key, file and transport options with a worked example. v0.1.1143

02 Bulk ChatGPT account import command NEW

90

Adds `magpie accounts import <codex|claude|antigravity|factory> <file>... [--yes]` to bulk-import ChatGPT accounts from Cockpit Tools' and Sub2API's exports, accepting multiple files at once.

Import a batch of ChatGPT accounts exported from Cockpit Tools or Sub2API without being prompted to confirm each one.

```
$ magpie accounts import codex export1.json export2.json --yes
```

— Exact runnable command and example provided. v0.1.1152

---

03 **Gateway conversation recording and transcript retrieval** NEW

90

Adds `SetGatewayRecording` and `SaveGatewayTurn` to enable opt-in local recording of gateway conversations, stored under `gateway-conversations/` in the config directory, and `GatewayTranscript` to retrieve the most recent captured turns (up to `gatewayReadMax` = 128) for a named agent/session pair, with a `Cut` flag indicating truncation. The sessions list now includes these gateway sessions, and `GET /api/sessions/manage?agent=<agent>` returns a `recorded` field so the sessions page shows the delete control only when conversation text is actually kept — replacing the inline recording controls bar with a dedicated settings card above the sessions list showing what is kept, where it is stored, and a labelled delete button.

Check whether any gateway conversation text is currently kept for an agent before deciding whether to offer a delete action.

```
$ curl -s 'http://localhost:<port>/api/sessions/manage?
agent=opencode' | jq '{recording, recorded}'
```

— Names every function, field, endpoint and limit involved, with a runnable check.

v0.1.1146 v0.1.1150

---

04 **DeepSeek Harness environment-variable expressions** NEW

90

Adds `!!js` expression support for environment variable references in DeepSeek Harness (`fmtDsh`), allowing secrets such as `GITHUB_TOKEN: !!js process.env.GITHUB_TOKEN` to be injected via the YAML `!!js` tag rather than plain string interpolation.

Inject a secret into a DeepSeek Harness MCP server without writing the credential into the config file, using the `!!js` YAML tag to read from the process environment at activation time.

```yaml
# $DSH_HOME/profiles/web/cordis.patch.yml
- id: mcp-client
  config:
    servers:
      github:
        command: npx
        args: ['-y', '@modelcontextprotocol/server-github']
        env:
          GITHUB_TOKEN: !!js process.env.GITHUB_TOKEN
```

— Exact YAML tag and config file shown with example. v0.1.1143

05 **Snow CLI agent integration** NEW 85

Adds Snow CLI agent support: magpie writes a `profiles/magpie.json` profile to `~/.snow` (or `$SNOW_CONFIG_DIR`), sets it as the active profile via `active-profile.json`, and restores the original on exit — Snow App's Sync Snow CLI API config picks it up automatically.

— Exact config paths and filenames given, though no CLI command shown. v0.1.1154

06 **Model reply limit exposure across API and CLI** NEW 85

Exposes a model's reply limit alongside its context window via `magpie model output`, `/v1/models`, Cursor's and Muse's model lists, and the Gateway page.

Check a Grok model's output token limit from the CLI before sizing a long-context task.

```
$ magpie model output
```

Query the reply limit for all models via the gateway API — useful for tooling that auto-selects models by their output cap.

```
$ curl http://localhost:<port>/v1/models
```

— Runnable CLI command and API endpoint given with examples. v0.1.1144

07 **Plugin management and marketplace improvements** IMPROVED 83

Adds `magpie plugin update` to bring a pinned plugin to npm's latest version, clearing the update prompt; orders plugin pre-release versions as semver pre-releases in both Go and the UI, showing the update on a plugin's row once its pre-release gets a

stable release; adds icon display for third-party plugins tagged `magpie-plugin`, reading the icon path from the package's `package.json`; gives provider and middleware entries their own distinct hue in the plugins card view; and, when two plugins sign in to the same provider, lists both with a per-row label indicating which one serves it and a Use for picker to switch between them.

Update a plugin that was pinned to an older version to bring it to npm's latest and clear the stale update prompt.

```
$ magpie plugin update
```

— Names every plugin UI and CLI surface added across releases. v0.1.1151 v0.1.1144 v0.1.1143

08 **Volcengine Ark usage window tracking** NEW 75

Reads Volcengine Ark Coding Plan and Agent Plan usage windows (5-hour, weekly, and monthly) via `GetCodingPlanUsage` and `GetAFPUsage` on `open.volcengineapi.com`, authenticated with the account's `AccessKeyID` and `SecretAccessKey`.

— Names exact API calls and host but no CLI step. v0.1.1142

09 **Deprecated model suppression from routing** NEW 73

When a provider returns a deprecation response, magpie removes that model from `/v1/models`, agent model lists, and routing groups for that provider for 7 days (`retiredFor`), then re-evaluates automatically.

— Names the endpoint, field and duration; behavior is automatic. v0.1.1148

10 **Reasoning state preservation across API translations** IMPROVED 68

Preserves OpenAI Responses API reasoning items — `id`, `summary`, and `encrypted_content` — through Responses-to-Responses translation in both directions, and preserves Gemini `thoughtSignature` on the signed function call and text part through translation, returning the original signature instead of a skip value.

— Names exact fields preserved but behavior is automatic. v0.1.1155

11 **Per-category personal-data redaction switches** NEW 68

Adds new redaction categories — SSNs, passports, IBANs, birthdays, MAC addresses, and serial numbers (on by default); home-folder names, `user@host`, public IP addresses, and cloud bucket names (off by default) — each individually switchable.

— Lists every category and default state but not the toggle's name. v0.1.1153

12 **Pi MCP config path alignment** IMPROVED 68

Pi MCP servers now write to `~/.pi/agent/mcp.json` (the file Pi 0.99+ loads) unless `pi-mcp-adapter` is loaded, preventing stale adapter configs from being sent.

– Names the exact file path and condition. v0.1.1151

13 **magpie-models.json key preservation and reasoning summaries** IMPROVED 68

`magpie-models.json` now preserves user-added keys on a model entry and declares `supports_reasoning_summaries`, enabling Codex before 0.145 to send effort and display thinking.

– Names the file and field but no edit example. v0.1.1151

14 **Grok trailing .0 stripping in tool-call arguments** IMPROVED 68

Strips trailing `.0` from whole-number tool-call arguments in Grok model replies on every route and protocol — so Codex and other strict-typed clients no longer reject calls like `2500.0` for `u64` / `i32` fields.

– Specific example and types given but fix is automatic. v0.1.1144

15 **service_tier forwarding to custom providers** NEW 68

Forwards the `service_tier` field (including `priority`, `ultrafast`, and `flex` tiers) that Codex requests to providers added by address, over both the Responses API and Chat API.

– Names the field, tier values and APIs but no setup example. v0.1.1138

16 **DNS rebinding protection in the gateway** NEW 65

The gateway now refuses loopback requests made under a hostname that isn't this computer's (e.g. a website whose DNS resolves to 127.0.0.1) with HTTP 403, unless an enabled gateway key is present; agents calling via `localhost` or `127.0.0.1` are unaffected.

– Explains the mechanism and status code but no config step for users. v0.1.1157

17 **Remote magpie decision and retrieval APIs** NEW 65

Adds remote magpie decision and retrieval APIs so a locally forwarded `/v1/systemone` request can reach decision providers on a remote gateway, with discovery of explicitly marked decision models including Clef and custom names.

– Names the endpoint and example models but no setup steps. v0.1.1145

18 **Loop guard for runaway model replies** NEW 65

Adds a loop guard (`loopGuard` / `holdWriter.cutLoop`) in the gateway that detects when a model's reasoning or text output has become a repeating cycle and ends the reply with an actionable error rather than letting it run to the token limit.

— Names the internal mechanism but gives no user-facing control. v0.1.1142

19 **Gateway remote-access key requirement for MAGPIE_ADDR**

IMPROVED 60

A gateway `MAGPIE_ADDR` now requires an enabled gateway key from callers on other machines (including container hosts), matching the sharing authentication model.

— Names the env var but not how to enable the key. v0.1.1151

THINNER COVERAGE BELOW

20 **Skill-change detection on library open** NEW 55

Skills from GitHub are checked for local edits as the skills page opens; a changed skill is flagged and prompts the user before an update overwrites it.

— Names the UI trigger but no config key. v0.1.1151

21 **Cursor HTTP/1.1 fallback on blocked networks** IMPROVED 55

Cursor users on a network that blocks HTTP/2 ('h2 is not supported') now fall back automatically to HTTP/1.1 via RunSSE + BidiAppend on api2.

— Automatic fix; nothing for the user to configure. v0.1.1151

22 **Codex failover routing visibility** IMPROVED 55

When Codex fails over through magpie to the user's other ChatGPT accounts, the Agents row now shows that routing and what disables it, and Codex's model list stays limited to its own models rather than merging with magpie's full list.

— Names the UI location but no config toggle. v0.1.1138

23 **Model-pause scheduling in routing groups** NEW 50

Scheduling rules can now pause a model within a routing group at set hours, days, or agents, removing it from rotation during those windows without fallback or affinity retention.

— Describes the rule but no command or config key to set it. v0.1.1155

24 **Video generation support across providers** NEW 50

Expands `generate_video` and the Magpie Image card to name every video model the gateway supports, not only Grok subscriptions, and adds OpenAI Videos API support for custom providers.

— Names the surfaces but no config key or command. v0.1.1152 v0.1.1146

25 **Session folder pagination** IMPROVED 50

Session folders now load only the latest 100 sessions with a Show more for pagination, and opening a session fetches only that session's data.

— States the limit but no config to change it. v0.1.1151

26 **Routing tab and provider editor UI refinements** IMPROVED 50

Provider editors retain a Fetch models again button after an initial model list is fetched, allowing re-fetch when an API key moves to another group; the Routing tab and provider editor also now show the saved routing choice and provider selection in agents' lists before Settings is opened.

— Names the UI locations but no config key or command. v0.1.1151 v0.1.1144

27 **Codex/ChatGPT account management refinements** IMPROVED 50

Supports running Codex subagents on a model set in the subagent's own row or on a ChatGPT account's model; distinguishes two Team seats sharing the same email address by the name each seat is saved under, so their usage is tracked separately; and automatically skips ChatGPT accounts that are at or over their usage limit (e.g. workspace owner usage limit reached) rather than switching to them.

— Names the behaviors but no settings or commands to configure them. v0.1.1142 v0.1.1138

28 **Bun SOCKS proxy HTTP bridge for plugins** IMPROVED 50

Passes bun a local HTTP bridge in front of a SOCKS system proxy so that plugin installs, updates, and removals succeed when only a SOCKS proxy is configured (previously failed with `UnsupportedProxyProtocol`).

— Fixes a named error automatically; nothing to configure. v0.1.1139

29 **magpie-image binary path fix** IMPROVED 40

Updates `magpie-image` to run the current computer's magpie binary regardless of the path saved by another computer or an older install.

— Describes the fix but gives no verification step. v0.1.1144

30 **Instant availability of new Claude models** IMPROVED 35

New Claude models listed by models.dev (e.g. Haiku 5.5) are available immediately after the account's last fetch, without waiting for a manual Refresh.

— States the behavior but gives no action for the user. v0.1.1150

31 **Multi-provider model ID routing order** IMPROVED 35

A bare model ID served by multiple providers now routes by the declared Providers order, with the behavior documented and tested.

— States the rule but gives no example of declaring order. v0.1.1148

32 **Quota-unavailability messaging detail** IMPROVED 35

Shows specific reasons — not shared there , key not taken , or not reached — on a remote computer's quota card instead of the generic Allowance unavailable message.

— Lists the new messages but no action beyond reading them. v0.1.1139

33 **Auto-mode server review scoped to Claude** IMPROVED 30

Scopes auto-mode server review to Claude only, with the verdict returned on every route.

— Thin one-line scope change with no mechanism detail. v0.1.1145

34 **Resumed reply fix for non-prefill models** IMPROVED 25

Ensures a resumed reply to a model that does not support prefill is answered correctly rather than refused.

— Bare bug-fix description, no named model or mechanism. v0.1.1145

WAS THIS USEFUL?  

Diagram Design

Feedback ⓘ changes since 2026-09-18 CODE ↗

RANK

Diagram Design is a Claude Code skill that generates editorial HTML and SVG diagrams and imports draw.io or Mermaid sources.

> Diagram Design shipped its full 44-type editorial diagram library (plus three new layouts — Architecture delta, Exploded axonometric, and Axonometric plan), a draw.io/Mermaid/Excalidraw import pipeline with named sizing and fidelity controls, new export-script flags for registry metadata and offline PNG, and cross-agent install/update paths with slash commands across five agent hosts.

└──▷ **WHAT SHIPPED · 10 FEATURES** most completely described first ? what's the number?

01 Font and brand token controls in style-guide.md NEW

85

Set `font source` to `system` in `skills/diagram-design/references/style-guide.md` to eliminate all network requests, falling back to platform serif, sans, and monospace faces. Edit brand tokens directly in the same file — semantic role names such as `accent`, `ink`, and `paper` propagate to every diagram, annotation, and the gallery, e.g. overriding `accent: '#0057FF'`, `ink: '#1A1A1A'`, `paper: '#FFFFFF'` by hand instead of running the onboarding prompt.

Switch to system fonts so generated diagrams make zero network requests — useful for air-gapped or privacy-sensitive environments.

```markdown
font source: system
```

Override brand tokens by hand so every diagram renders in your own palette and typography without running the onboarding prompt.

```markdown
# Edit the token table at the top of style-guide.md:
# accent: '#0057FF'
# ink:    '#1A1A1A'
# paper:  '#FFFFFF'
```

— Exact config file, keys, and example override values given product docs

02 draw.io, Mermaid and Excalidraw import and redraw pipeline

NEW

85

Supports import and redraw of draw.io files in `.drawio`, `.drawio.xml`, `.drawio.png` (with embedded diagram), and `.drawio.svg` formats including compressed base64 payloads, plus Mermaid and Excalidraw source files, into the Diagram Design system. Output can target named sizes — `doc-inline`, `doc-wide`, `slide-16x9`, `slide-4x3`, `social-og`, `social-square`, `print-a4-landscape`, `print-a3-landscape`, `print-letter-landscape`, and `fit` — and named detail levels — `faithful` (up to 24 nodes), `balanced` (up to 12), `simplified` (up to 7) — applied via a fixed degrade ladder, producing a fidelity ledger at the end of every import listing what was merged, collapsed, or dropped.

— Every format, size target, and detail-level name given; no exact command shown

product docs

---

03 **Export script flags for registry metadata and offline PNG**

NEW 85

Adds `--registry` flag to the export script to emit `<basename>.registry.json` containing structured `data-block-*` metadata for every block in a traceable block decomposition diagram; adds `--system-fonts` flag to the SVG export path to skip the Google Fonts import and produce a fully offline SVG; and supports PNG export via Playwright at 2× resolution by default (`pip install playwright && playwright install chromium`).

— Named flags, output file pattern, and a runnable install command product docs

---

04 **Cross-agent installation and update paths** NEW 80

Install the skill into any agent root with `npx skills add cathrynlavery/diagram-design`, and pull updates to a standalone install with `npx skills update diagram-design`. In Claude Code, enable auto-update via `/plugin` → Marketplaces → diagram-design → Enable auto-update, then reload with `/reload-plugins`. For OpenCode, copy or symlink `skills/diagram-design/` to `.opencode/skills/diagram-design` (per-project) or `~/.config/opencode/skills/diagram-design` (global).

Install the skill into every agent root you use (Cursor, Claude Code, Codex, etc.) so any agent can generate editorial diagrams immediately.

```
$ npx skills add cathrynlavery/diagram-design
```

Pull the latest skill updates into an existing standalone install without reinstalling from a marketplace.

```
$ npx skills update diagram-design
```

— Runnable commands and exact UI navigation for each host `product docs`

---

05 **Marketplace reinstall required for continued auto-updates** 75

`BREAKING`

An existing standalone `npx skills add` copy will not follow the Codex marketplace automatically — remove the standalone copy and reinstall from the Codex marketplace to receive future updates. A personal Cowork copy must likewise be uninstalled and Diagram Design reinstalled from the organization marketplace to receive future version bumps through the native update path.

— States exact before/after and the migration steps needed `product docs`

---

06 **Architecture delta diagram type** NEW 65

New Architecture delta diagram type compares topologies through a Before · Changes · After ledger of added, removed, changed, moved, and rewired objects.

— Clear mechanism (ledger categories) but no invocation example `product docs`

---

07 **Exploded axonometric diagram type** NEW 65

New Exploded axonometric diagram type renders objects in 2:1 dimetric projection with parts lifted apart at equal gaps, including optional animation that assembles then explodes once.

— Concrete projection ratio and animation behaviour given, no usage example `product docs`

---

08 **Slash commands via native marketplace packages** NEW 65

Adds `/export-diagram`, `/import-mermaid`, `/profile`, and `/doctor` slash commands via native marketplace packages for Claude Code, Codex, Factory Droid, Pi, and GitHub Copilot.

— Four commands and five hosts named, but no behaviour detail per command `product docs`

---

09 **44 editorial diagram types with light/dark/full variants** NEW 63

Generates 44 diagram types as self-contained HTML+SVG files — each in minimal light, minimal dark, and full-editorial variants — with no build step, JavaScript, or external image dependencies.

— Names the variant system and zero-dependency output but no invocation syntax

`product docs`

THINNER COVERAGE BELOW

---

10 **Axonometric plan diagram type** NEW 55

New Axonometric plan diagram type renders floor plans and campus sites in the same dimetric projection, with walls cut at desk height.

— Names the technique but thinner mechanism than its siblings product docs

└──▷ **BREAKING ON UPGRADE**

! An existing standalone `npx skills add` copy will not follow the Codex marketplace automatically — remove the standalone copy and reinstall from the Codex marketplace to receive future updates.

! A personal Cowork copy must be uninstalled and Diagram Design reinstalled from the organization marketplace to receive future version bumps through the native update path.

! An existing standalone `npx skills add` copy will not follow the Codex marketplace automatically — remove the standalone copy and reinstall via the Codex marketplace to receive future updates.

! A personal Cowork installation must be uninstalled and reinstalled from the organization marketplace to receive future version bumps through the native update path.

changes since 2026-09-18 CODE ↗

Diagram Design adds heatmap, architecture delta, exploded axonometric, and axonometric plan diagram types, plus a print-a3-landscape size preset.

└──▷ **TRY IT**

Rebuild all exploded axonometric examples from the 2:1 dimetric projection after changing the model.

```
$ python3 scripts/build-exploded-examples.py
```

Verify that no generated exploded axonometric example file is stale after a model change.

```
$ python3 scripts/build-exploded-examples.py --check
```

Adds `print-a3-landscape` size preset to the output spec for print-ready diagram exports.

Adds heatmap as a visual type — a rectangular grid where fill opacity encodes a quantitative value across two categorical axes.

Adds architecture delta diagrams as a new diagram type.

Adds the exploded axonometric diagram type, with coordinates generated via a 2:1 dimetric iso(x, y, z) projection.

Prepares the Codex manifest for OpenAI's plugin directory.

+1 more

WAS THIS USEFUL?  

OpenAI Codex CLI

Feedback ⓘ rust-v0.163.0-alpha.5 NOTES ↗

CODE ↗ RANK

OpenAI Codex CLI runs an agent in the terminal that reads, changes, and tests code in local repositories.

> Codex CLI's alpha.5 release adds clipboard forwarding to Herdr terminals via OSC 52, exposes board permission data through the agent API, and introduces structured permission-denied errors, alongside several smaller internal and configuration refinements.

└──▷ **WHAT SHIPPED · 9 FEATURES** most completely described first ? what's the number?

01 **Clipboard forwarding to Herdr via OSC 52** NEW 80

Detects Herdr sessions via `HERDR_ENV=1` and forwards clipboard copies (`/copy` and `Ctrl+O`) to the terminal using OSC 52, even when native copying succeeds.

— Names env var, shortcuts and protocol; mechanism explained. rust-v0.163.0-alpha.5

02 **Board permissions exposed in get_channels API** NEW 75

Exposes optional `board_permissions` and per-channel `permissions` fields in `get_channels` responses, surfacing effective remote board permissions to agents.

— Names exact API fields and endpoint response. rust-v0.163.0-alpha.5

03 **Structured permission_denied tool errors** NEW 75

Surfaces HTTP 403 `permission_denied` responses as structured tool errors with a message and optional `action` and `resource` fields, truncating oversized details to fit the output budget.

— Describes error shape and fields but no direct user action. rust-v0.163.0-alpha.5

THINNER COVERAGE BELOW

04 **Per-turn Cyber access forwarded to Guardian** NEW 30

Forwards per-turn Cyber access programs to Guardian.

— Names Guardian but mechanism unclear. rust-v0.163.0-alpha.5

05 **Timeout-aware exec-server configuration reads** IMPROVED 30

Adds timeout-aware exec-server configuration reads.

— Names exec-server but no flag or timeout value. rust-v0.163.0-alpha.5

06 **Reserved Serde JSON key rejection in code mode** BREAKING

30

Rejects reserved Serde JSON keys in code mode.

— Names Serde and code mode but no key list or migration note. rust-v0.163.0-alpha.5

07 **Opt-in retention annotations for tool outputs** NEW

25

Adds opt-in retention annotations for submitted and per-turn tool outputs.

— Bare description, no mechanism or config name given. rust-v0.163.0-alpha.5

08 **Experimental thread item lookup by ID** NEW

25

Adds experimental thread item lookup by ID.

— Thin mention, no interface or scope detail. rust-v0.163.0-alpha.5

09 **Capability roots refresh from owner config** IMPROVED

25

Refreshes persisted capability roots from owner-provided configuration.

— No config key or trigger named. rust-v0.163.0-alpha.5

WAS THIS USEFUL? 👍 👎

Sourcegraph Amp

Feedback ⓘ

Use Your Claude Plan in Amp (2026-10-10) BLOG ↗ RANK

Amp is Sourcegraph's agentic coding tool for the terminal and editor, running multi-step edits with subagents and shared team threads.

> Amp's headline addition is a Claude Code mode that lets threads run on a user's own Claude Pro/Max subscription with full Amp feature parity, alongside ChatGPT subscription sign-in, cross-workspace thread search, and smaller runner/CLI improvements.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 **Claude Code mode using your own Claude plan** NEW 88

Adds Claude Code mode (selectable via `Mode > Claude Code` when creating a new thread) that routes threads through the Claude Agent SDK using a linked Claude Pro/Max plan instead of Amp's own agent, available free to Claude Pro/Max subscribers. On self-hosted runners, Amp uses the runner machine's own `claude` install, Claude login, and `~/.claude` settings, stripping `ANTHROPIC_API_KEY`, `ANTHROPIC_AUTH_TOKEN`, and Bedrock/Vertex/Foundry env vars so only the plan is used; the `IS_SANDBOX=1` environment variable lets Claude Code start on Linux/macOS containers that must run as root. Full Amp capabilities — orbs, runners, thread sharing, portals, multiplayer, and thread-to-thread messaging and orchestration — extend to threads running in this mode.

— Mechanism, env vars and UI path named, no exact CLI command

Use Your Claude Plan in Amp (2026-10-10)

02 **amp update command for runners** NEW 70

`amp update` immediately updates the Amp CLI on a self-hosted runner without waiting for the automatic self-update, ensuring Claude Code threads run the required current version.

Update the Amp CLI on a self-hosted runner immediately so Claude Code threads have the required current version.

```
$ amp update
```

— Runnable command, minimal surrounding detail product docs

03 **Suppress Painter tool via AGENTS.md** IMPROVED 70

Painter (image generation) costs can be suppressed by adding `Do not use the painter tool.` to `AGENTS.md` (or `CLAUDE.md` if present).

Prevent the Painter tool from consuming Amp credits on Claude Code threads by opting out in your agent instructions file.

```
$ echo 'Do not use the painter tool.' >> AGENTS.md
```

— Exact config text and file path given `product docs`

THINNER COVERAGE BELOW

04 **Thread search across workspace** NEW 45

Amp can now search past threads — your own and workspace members' — by keyword, repository, author, date, or task.

— Named filters but no command or UI path given `product docs`

WAS THIS USEFUL? 👍 👎

Letta Code

Feedback ⓘ v0.34.10 NOTES ↗ RANK

Letta Code is a terminal coding agent with persistent memory and identity across development sessions.

> Letta Code shipped a listener scoping flag for its server command and added MCP OAuth authentication support in the browser-based Chat interface.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Org-scoped server listener flag** NEW 60

The `letta server` listener command gains an `--org` flag to scope a listener to a single organization.

— Names exact flag and command, no further mechanism given v0.34.10

THINNER COVERAGE BELOW

02 **MCP OAuth in Chat interface** NEW 45

Adds an MCP OAuth authentication flow directly from the Chat interface at `chat.letta.com` for browser/device use.

— Names interface and flow but no steps or endpoints v0.34.9

WAS THIS USEFUL? 👍 👎

Anthropic Claude Code

Feedback ⓘ v2.1.296 NOTES ↗

CODE ↗ RANK

Claude Code is Anthropic's terminal coding agent that plans, edits, and tests code in local repositories.

> Claude Code v2.1.296 extends gateway policy controls to Claude Desktop, adds per-subagent compaction control and new tuning environment variables, raises MCP description limits, and ships HIPAA-ready managed-settings examples alongside admin UI improvements for session filtering and team memory management.

└──▷ **WHAT SHIPPED · 9 FEATURES** most completely described first ? what's the number?

01 `code` policy key in Claude apps gateway NEW

83

Adds a `code` key to the Claude apps gateway's `managed.policies[]`, applying the same settings as `cli` to Claude Desktop's Code tab and enabling Claude Desktop's gateway mode.

Apply the same gateway policy to Claude Desktop's Code tab as to the CLI — and turn on Desktop gateway mode — by adding the `code` key alongside `cli` in managed settings.

```json
{
  "managed": {
    "policies": [
      {
        "cli": { "allowedModels": ["claude-sonnet-4-5"] },
        "code": { "allowedModels": ["claude-sonnet-4-5"] },
        "desktop": {}
      }
    ]
  }
}
```

— Exact config key with runnable JSON example v2.1.296

02 New environment variables for model and retry tuning NEW

73

Adds `CLAUDE_CODE_WORKFLOW_SUBAGENT_MODEL` to run every workflow agent on a single specified model while other subagents keep their own model assignments, and `CLAUDE_CODE_OVERLOADED_RETRY_MAX_DELAY_MS` to set a longer maximum backoff delay when retrying overloaded (529) requests.

— Two named env vars grouped, no example shown v2.1.296

03 **Per-subagent compaction window control** NEW 70

Adds `autoCompactWindow` to subagent frontmatter and `--agents` definitions so a subagent can trigger auto-compaction earlier than the main conversation's context window.

— Names exact field but gives no usage example v2.1.296

04 **HIPAA managed-settings example configs** NEW 70

Adds HIPAA managed-settings example files — `settings-hipaa.json`, `managed-mcp-hipaa.json`, and `README-hipaa.md` — in `examples/settings/` as a starting-point configuration for organizations limiting how session content leaves a developer's machine.

— Named files and path, no live example shown v2.1.296

05 **`allow_large` option for Read tool** NEW 68

Adds `allow_large` option to the Read tool so Claude can read a text file past the usual size limits in a single call when the full file is needed and context allows.

— Named flag described but no example call v2.1.296

THINNER COVERAGE BELOW

06 **MCP tool description character limit raised** IMPROVED 55

Changes the default character limit on MCP tool descriptions and MCP server instructions from 2,048 to 4,096 characters.

— Exact before/after numbers, no action required v2.1.296

07 **Status filter for Sessions and Runners lists** NEW 53

Adds a status filter to the Sessions and Runners lists on a self-hosted environment's Activity tab in admin settings, allowing multiple statuses to be selected at once.

— UI path given but no specific status values v2.1.296

08 **Memory file management in Activity page** NEW 53

Adds creating, editing, and deleting workspace and channel memory files on the Activity page's Memory tab for members with the Claude Tag Admin permission.

— UI path and permission named, little else v2.1.296

09 **Hook-conflict notice in /plugin** IMPROVED 40

Adds a note in `/plugin` when a plugin's hooks are omitted because another enabled plugin shares the same name.

— Thin one-line UI notice, minimal mechanism given v2.1.296

WAS THIS USEFUL? 👍 👎

sirmalloc ccstatusline Feedback ⓘ v2.2.31 NOTES ↗

RANK

A customizable statusline for Claude Code CLI with powerline support, themes, and widget-based configuration.

> ccstatusline's v2.2.31 release adds two new statusline widgets for cost tracking, inline label editing in the TUI, and faster piped rendering.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 Session Cost Rate and Extra Usage Daily Budget widgets NEW

55

A new Session Cost Rate widget tracks session cost per hour of active API time or total session time, with a one-minute minimum before displaying a rate. A new Extra Usage Daily Budget widget spreads the remaining monthly spending limit over remaining UTC days, with an option to skip future weekend days.

— Explains mechanism of both widgets but no config key named v2.2.31

02 Editable widget label prefixes via `e` key IMPROVED

50

Pressing `e` in the TUI lets you edit a widget's label prefix directly.

— Exact keypress given but limited explanation of scope v2.2.31

03 Faster piped status-line rendering IMPROVED

40

Piped status-line renders now skip loading the interactive TUI, reducing startup work on every repaint.

— States mechanism but no measured improvement or numbers v2.2.31

WAS THIS USEFUL? 👍 👎

GitHub Copilot CLI

Feedback ⓘ v1.0.95 NOTES ↗ RANK

GitHub Copilot CLI is a terminal-based coding agent that can plan, write, and test code in local repositories.

> GitHub Copilot CLI v1.0.95 adds native Entra broker authentication on macOS, new sandbox credential config keys, and fixes context handling in ACP sessions.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 **Sandbox credential config for host injection** NEW 70

Adds `injectHosts` keys under sandbox credential config to `copilot config`, with tab completion in Bash, Zsh, and Fish.

— Names exact config key and shells supported v1.0.95

02 **--context fix for ACP sessions** IMPROVED 70

`--context` now applies to new and resumed ACP sessions instead of silently falling back to the default or previously saved context tier.

Set a specific context tier when starting or resuming an ACP session so Copilot uses the right scope rather than a stale or default one.

```
$ copilot --context <context-tier>
```

— Includes before/after behavior and runnable example v1.0.95

THINNER COVERAGE BELOW

03 **Native Entra broker authentication on macOS** NEW 45

Adds native Microsoft Entra broker authentication on macOS when available, with automatic browser fallback.

— Describes behavior but no command or config shown v1.0.95

04 **Managed plugin setup retry behavior** IMPROVED 40

Managed plugin setup now retries hourly or after policy changes instead of on every message failure.

— Clear before/after but no actionable surface named v1.0.95

WAS THIS USEFUL? 👍 👎

DeepSeek Harness

Feedback ⓘ dsh-v0.2.1-alpha.2 CODE ↗

RANK

DeepSeek Harness is an open-source agent harness for building coding agents with plugins.

> DeepSeek Harness's alpha.2 release restores proper argument pass-through for the `dsh plugin` command and rounds out the client with independent font controls, microphone selection, a local bundle registry, and configuration-free macOS packaging.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 Pass-through fix for `dsh plugin` subcommand IMPROVED

80

Adds `-h, --help` option and passThroughOptions() to the `dsh plugin` subcommand so pnpm arguments are forwarded verbatim without Commander intercepting them.

Forward pnpm arguments to a named profile without Commander swallowing flags — the restored `-h` and pass-through mode make this work correctly.

```
$ dsh plugin --profile my-profile add some-package
dsh plugin --profile my-profile -h
```

— Names exact flag and method, includes runnable example dsh-v0.2.1-alpha.2

THINNER COVERAGE BELOW

02 Local loopback registry for Official bundles NEW

45

Serves locally built Official bundles from a loopback registry for offline or dev-time use.

— Explains purpose but no command or endpoint to invoke it dsh-v0.2.1-alpha.2

03 Independent font and size controls in client NEW

30

Enables independent configuration of text, code, and terminal fonts and sizes in the client.

— No config key or UI path given dsh-v0.2.1-alpha.2

04 Microphone selection in voice-input UI NEW

30

Adds microphone selection with a menu-only input-level display in the voice-input UI.

— Describes behavior but no exact navigation or settings path dsh-v0.2.1-alpha.2

05 **Configuration-free macOS desktop packaging** IMPROVED

20

Makes local macOS desktop packaging configuration-free.

— Bare claim with no detail on what changed `dsh-v0.2.1-alpha.2`

WAS THIS USEFUL? 👍 👎

Amazon Kiro

Feedback ⓘ

CLI: Memory Across Chat Sessions: In local V3 sessions, Kiro can learn useful context fr

BLOG ↗ RANK

Kiro is an agentic development environment that uses spec-driven workflows to plan, build, and maintain software.

> Kiro's biggest window addition is reusable multi-agent Workflows with persistent chat Memory, alongside tighter untrusted-workspace safeguards, enterprise sign-in controls, and more flexible Hook matching across both the CLI and IDE.

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

01 Multi-agent Workflows in chat NEW

88

Multi-agent Workflows can be enabled via `kiroAgent.workflows.enabled` in Settings (or Workspace Configuration in Agent Focus), running each step in its own agent session with pause, resume, and stop controls. Workflow recipes are stored in `.kiro/workflows/`, are protected from agent writes, and are not loaded in untrusted workspaces until trust is granted; a feedback survey link was added to the chat Workflows drawer in the desktop app (unavailable in GovCloud, AWS isolated Regions, and China Regions).

Opt in to multi-agent Workflows for a project so Kiro can run long, multi-step plans in the background while you keep working in the parent chat.

📍 In Settings, search for `kiroAgent.workflows.enabled` and set it to true, then open a new chat session to start using Workflows.

— Names config key, storage path, and controls; UI entry point not pinpointed

IDE: Workflow Feedback, Power Search, a… IDE 1.2.56: Workflows, Safer Untrusted …

02 Memory across chat sessions in CLI NEW

85

In local V3 sessions, Kiro can learn useful context from your work — such as repository commands and conventions — and recall it in later chat sessions, with automatic updates after sessions end. The `/memories` slash command configures access via `config` (`read & write`, `read only`, or `off`) and lets you `list` to search, preview, and open stored memories.

Set memory access to read-only in a V3 session so Kiro can use existing context without adding or changing memories — useful in shared or sensitive workspaces.

📍 Type `/memories` in the chat, select config, then choose `read only`.

— Names the exact subcommand, options, and gives usage steps

CLI: Memory Across Chat Sessions: In lo…

03 **Agent JSON migration command** NEW 80

`kiro-cli agent upgrade <dir>` upgrades top-level agent JSON files to the format both V2 and V3 load, with `--dry-run` to preview changes without writing and `--no-backup` to skip default backups.

Migrate agent JSON files in a directory to the V2/V3-compatible format, previewing what will change before committing.

```
$ kiro-cli agent upgrade ./agents --dry-run
```

— Runnable command with named flags and a dry-run example

CLI: Memory Across Chat Sessions: In lo…

04 **Category and selector Hook matchers in IDE** IMPROVED 75

PreToolUse and PostToolUse Hooks now support category and selector matchers — `@builtin`, `@mcp`, `@server`, `@server/tool` — replacing exhaustive per-tool-ID lists.

Match all MCP tools in a PreToolUse Hook without listing each tool ID individually.

📍 In your Hook configuration, set the matcher to `@mcp` under `PreToolUse` to intercept every MCP tool call in one selector instead of enumerating tool IDs.

— Names exact matcher syntax with a worked example

IDE: Workflow Feedback, Power Search, a…

05 **ACP V3 sign-in delegation** NEW 60

`kiro-cli acp` V3 sign-in defaults to your Kiro CLI login, with `--auth-method=client` to delegate sign-in to the connecting client instead.

Connect to an ACP server in V3 while letting the connecting client handle authentication instead of using your stored Kiro CLI login.

```
$ kiro-cli acp --auth-method=client
```

— Named command and flag with a runnable example CLI: Memory Across Chat Sessions: In lo…

THINNER COVERAGE BELOW

06 **Untrusted workspace trust prompts** NEW 50

Kiro now prompts before the agent modifies agent, Hook, or Power files, or any Kiro configuration — even when broader file permissions allow the write — and, in untrusted workspaces, prompts before each command runs even when previously allowed.

— Clear before/after behaviour but no config toggle given

IDE 1.2.56: Workflows, Safer Untrusted …

07 **Spec-session-only slash commands** IMPROVED 50

`/quick-spec`, `/architecture-selection`, and `/bug-fix` are now spec-session-only, and are removed from the command list in other session types.

— Names the commands and scoping change, no further mechanism

IDE: Workflow Feedback, Power Search, a…

08 **Hook group matching in V2 TUI** NEW 50

V2 TUI Hook groups let matchers such as `read` cover every tool in the group, so a blocking `preToolUse` Hook can stop any tool in the group; `tool_tags` was added to Tool Hook input so scripts can check a tool's group.

— Names matcher and field but no runnable example given

CLI: Memory Across Chat Sessions: In lo…

09 **MCP server startup and environment controls in V3** IMPROVED 50

Custom agents now start only the `mcp.json` servers allowed by their tool settings, and `stdio` MCP servers now receive the full agent environment via environment inheritance.

— Names the config file and server type, no setup steps shown

CLI: Memory Across Chat Sessions: In lo…

10 **Agent Focus Mode views for Workflows and Artifacts** NEW 45

Agent Focus Mode gains dedicated views for Workflows and Artifacts, showing step activity and artifact previews, with an Open in IDE action on every artifact.

— Describes new views but gives no exact navigation path

IDE 1.2.56: Workflows, Safer Untrusted …

11 **Per-profile device flow sign-in control** NEW 45

Administrators can turn off device flow sign-in per Kiro profile; the CLI hides those profiles during device flow login and profile switching while browser sign-in remains available.

— Explains admin control and CLI behaviour, no config key named

CLI: Memory Across Chat Sessions: In lo…

12 **Enterprise sign-in controls in IDE** NEW 40

Administrators can restrict which sign-in methods are permitted, prefill organization sign-in details, and add a help link to the sign-in screen.

— Lists admin capabilities without exact settings or UI path

IDE 1.2.56: Workflows, Safer Untrusted …

13 **Telemetry routed to enterprise profile's AWS Region** IMPROVED 30

Kiro now routes telemetry and activity data to the AWS Region of the selected enterprise profile.

— States the behaviour but no mechanism or configuration detail

IDE 1.2.56: Workflows, Safer Untrusted …

14 **Name filtering in Available Powers view** NEW 30

Adds name-based filtering in the Available Powers view to find a Power faster.

— Bare description of a small UI filter IDE: Workflow Feedback, Power Search, a…

WAS THIS USEFUL?  

AGENT

◆ MCP Servers & Tooling

gram

Feedback ⓘ server@2.21.0 NOTES ↗ RANK

Gram provides MCP servers and AI-agent tools from APIs with access controls and observability.

> Gram's biggest addition this window is a new Signals intelligence product area for building custom signals and sensors, backed by a shared OAuth authorization server and a `credential_owner: self` mode for remote session clients; the dashboard also picked up a redesigned Explore query builder, shared dashboard filters, and expanded analytics tooling.

└──▷ WHAT SHIPPED · 11 FEATURES most completely described first ? what's the number?

01 Shared credential OAuth mode for remote session clients NEW

88

Setting `credential_owner: self` on remote session client creation makes the client hold one upstream OAuth credential — using the `client_credentials` grant — shared across all callers instead of per-user credentials, and adds `json_web_key_set_id` support for `private_key_jwt` clients in a single call.

Create a service-account-style remote session client that shares one upstream credential across all callers using a private key JWT, no per-user connect step required.

```json
{
  "credential_owner": "self",
  "json_web_key_set_id": "<key-set-id>",
  "issuer": "https://auth.example.com"
}
```

— Exact field names and a runnable example payload given. server@2.21.0

02 Per-principal install modes for plugins NEW

86

`setPluginAssignments` now accepts `install_modes` per principal URN, and `agent.getPlugins` returns `install_mode`, supporting `required`, `default`, and `available` modes; when assignments overlap for a principal, the strictest mode wins.

Assign a plugin with per-principal install modes so admins get it required and regular users see it as opt-in.

```javascript
setPluginAssignments({
  "plugin_id": "<plugin-id>",
  "install_modes": {
    "urn:principal:org:admins": "required",
    "urn:principal:org:users": "available"
  }
})
```

— Named methods, modes, and resolution rule with a worked example. `server@2.21.0`

03 **Signals intelligence: custom signals and sensors** NEW

78

A `signals_intelligence` product feature, gated by an org-level flag, controls sidebar, command-palette visibility, and direct page access to new Signals intelligence configuration management. New project-scoped management APIs support creating custom signals and sensors with cursor pagination, project authorization, and transactional audit events; sensors support multi-label, exclusive-choice, and ordered-score modes, incomplete drafts, and atomic ordered-membership updates. Sensors and signals also gained project-scoped `slug` fields — accepted on creation and update, defaulted from display names, exposed in all resource responses, and reserved through soft-deletion — and these slugs can now be created and edited directly in their dashboard forms with validation for the shared slug format.

— Full field and mode names but no exact endpoint path given. `server@2.21.0` `dashboard@0.135.0`

04 **Analytics tooling expansion: new MCP tools, skills dataset, count_distinct**

NEW

77

Platform MCP gains `describe_analytics_catalog`, `list_analytics_dimension_values`, and `run_analytics_query` tools so agents can query the same datasets as the Explore UI. A new `skills` analytics dataset adds one row per skill invocation, enabling ranking and counting of skills in Explore, saved widgets, and dashboards. Analytics queries also gain `count_distinct` aggregation, letting Explore and saved widgets count distinct tools, servers, sessions, or people in tool-call queries.

— All tool and dataset names included, no example query shown. `server@2.21.0`

05 **Custom trusted-platform registration flows** NEW

67

`workloadIdentities.getCustomFlows` (requires `workload:read`) serves YAML-driven forms for registering and editing trusted platforms not already in the Access Hub

catalog.

— Named method and permission but no usage example. server@2.21.0

06 **Shared OAuth authorization server for MCP** NEW 64

A shared authorization server on the authentication host now serves metadata, token, and revocation endpoints, accepting the workload grant — including resource-less grants from Claude Tag — with issued tokens scoped to every MCP server the issuer is permitted to access.

— Describes endpoints and grant types but no exact paths given. server@2.21.0

07 **Shared filter bar across dashboards** NEW 64

Dashboards gain a shared filter bar offering date range and filterable fields (user, agent, model, MCP server, status), with Save filters and Reset filters controls.

— Named fields and controls, but only a UI path. dashboard@0.135.0

08 **Redesigned Explore query builder** IMPROVED 62

The Explore query builder is redesigned as a two-line Datadog-style editor with `field:value` pill filters, a sentence-form aggregation line, and chart controls moved into the results header.

— Describes new UI layout, no config surface named. dashboard@0.135.0

THINNER COVERAGE BELOW

09 **Context windowing for prompt-attachment risk events** IMPROVED 54

Risk events found in prompt attachments (such as spreadsheets) now show context by windowing around the prompt the attachment was sent with, and the finding drawer displays the flagged attachment beneath that prompt.

— Explains mechanism but no UI path or config given. dashboard@0.135.0

10 **Add-to-dashboard action on widgets** NEW 48

An Add to dashboard action on Widgets tab rows and cards lets a widget be placed on any editable dashboard, or used to start a new one, from wherever it appears.

— Clear UI action but no further mechanism detail. dashboard@0.135.0

11 **Persistent search term on MCP page** IMPROVED 35

The MCP page now keeps the current search term in the URL, so navigating back preserves the search state.

— Minor UX fix with little further detail. dashboard@0.135.0

WAS THIS USEFUL? 👍 👎

◆ AI Agent Frameworks

CrewAI

Feedback ⓘ 1.15.27 NOTES ↗ RANK

Open-source Python framework for building multi-agent workflows using role-based AI agent Crews and event-driven Flows.

> CrewAI 1.15.27 adds DeepInfra as a new OpenAI-compatible LLM provider and XPU hardware acceleration, introduces per-run cost and time tracking for evaluation runs, and improves pre-flight observability for deployments.

└──▷ **WHAT SHIPPED · 4 FEATURES** most completely described first ? what's the number?

01 Per-run cost and time tracking in eval — IMPROVED — 70

Implements per-run cost and time tracking in `crewai eval --models`, surfacing spend and latency for each evaluation run instead of repeated progress lines.

– Names exact command flag and explains the before/after behaviour 1.15.27

THINNER COVERAGE BELOW

02 DeepInfra provider support — NEW — 45

Adds `deepinfra` as an OpenAI-compatible provider, enabling DeepInfra-hosted models to be used without writing a custom wrapper.

– Names the provider but no config key or code example shown 1.15.27

03 XPU device support in OpenCLIP provider — IMPROVED — 40

Adds `XPU` to the OpenCLIP provider's device options, broadening hardware acceleration support.

– Names the device option but no usage detail given 1.15.27

04 Deployment pre-flight observability — IMPROVED — 20

Records what the run app's Deploy step encountered before an attempt, improving observability for deployment pre-flight checks.

– Vague description with no named surface or mechanism 1.15.27

WAS THIS USEFUL? 👍 👎

Feynman

Feedback ⓘ v0.5.33 NOTES ↗ RANK

Feynman is a personal AI research assistant that searches papers and the web, builds literature reviews, and produces cited research reports.

> Feynman's pi-web-access component gained a new paid search provider and broader model compatibility for key-free authentication.

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

01 **Ceramic search provider and key-free model support in pi-web-access**

NEW 50

pi-web-access 0.38.0 adds Ceramic as a new paid web search provider, selectable on demand, and adds `fetch_content` answer mode support for models whose provider authenticates without an API key. The release also fixes proxy, key-free, and model compatibility issues.

— Names provider and mode but no setup steps or config keys given v0.5.33

WAS THIS USEFUL? 👍 👎

LiveKit Agents

Feedback ⓘ livekit-agents@1.8.6 NOTES ↗

LiveKit Agents builds realtime voice and video AI agents on LiveKit's WebRTC infrastructure, with turn detection, interruption handling, and telephony.

Speechify TTS plugin gains streaming synthesize() and prewarm() with HTTP connection reuse.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch livekit-agents@1.8.6
https://github.com/livekit/agents.git
# already have the repo? check out this version:
$ git checkout livekit-agents@1.8.6
```

NEW INTEGRATION Nabrah Arabic STT plugin

UPDATED INTEGRATION Sarvam TTS — bulbul:v4-flash model added

UPDATED INTEGRATION Cerebras — qwen-3.8-27b added to CerebrasChatModels

WAS THIS USEFUL? 👍 👎

CopilotKit

Feedback ⓘ v1.78.0 NOTES ↗ RANK

CopilotKit connects AI agents to application interfaces with shared state, tools, and generative UI components.

> CopilotKit v1.78.0 changes how the runtime picks the OpenAI API surface for proxied requests and switches on automatic trajectory capture for learning objects, both with explicit opt-outs.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 OpenAI API surface routing and override

BREAKING 85

When `OPENAI_BASE_URL` points at a non-OpenAI/non-Azure host, `'openai/...'` model strings now route through the Chat Completions API instead of the Responses API, which breaks Responses-only features like reasoning summaries when going through proxies such as LiteLLM, Portkey, or Helicone. The new `COPILOTKIT_OPENAI_API` environment variable (e.g. set to `responses`) lets you pin the API surface explicitly to restore the previous behaviour, and the runtime logs a warning when it falls back to Chat Completions.

Prevent the runtime from silently downgrading to Chat Completions when routing through an OpenAI-compatible proxy that relies on Responses API features like reasoning summaries.

```
$ COPILOTKIT_OPENAI_API=responses node server.js
```

— Names env var, values, affected proxies, and exact before/after behaviour. v1.78.0

02 Automatic trajectory capture for learning objects

BREAKING

77

A `learning` object without a `trajectoryId` now automatically starts trajectory capture on mount instead of requiring a manual startTrajectory() call. Set `autoStart: false` on the `learning` prop (e.g. `learning={{ onError, autoStart: false }}`) to keep the previous manual-start behaviour.

Opt out of automatic trajectory capture so you can call startTrajectory() yourself at the right moment in your workflow.

jsx

```jsx
learning={{ onError: handleError, autoStart: false }}
```

— Names exact prop and opt-out flag with before/after behaviour. v1.78.0

└──▷ BREAKING ON UPGRADE

! With `OPENAI_BASE_URL` set to a non-OpenAI/non-Azure host, `'openai/...'` model strings now use the Chat Completions API instead of the Responses API; features only available on the Responses API (e.g., reasoning summaries) stop working through proxies like LiteLLM, Portkey, or Helicone — set `COPILOTKIT_OPENAI_API=responses` to restore previous behaviour.

! A `learning` object without `trajectoryId` now automatically starts trajectory capture after mount; add `autoStart: false` (e.g., `learning={{ onError, autoStart: false }}`) to keep calling startTrajectory() yourself.

WAS THIS USEFUL? 👍 👎

PydanticAI

Feedback ⓘ v2.55.0 NOTES ↗ RANK

PydanticAI is a framework that builds type-safe Python agents with dependency injection, model integrations, tools, and structured outputs.

> PydanticAI v2.55.0 adds Conversation persistence with hosted Postgres stores, cross-provider prompt caching controls, new WebRTC/GPT-Live call controls, and an OpenAIDecisionsModel backend, alongside a sizable set of breaking changes including dropping Python 3.10 support.

└──▷ **WHAT SHIPPED · 12 FEATURES** most completely described first ? what's the number?

01 Cross-provider prompt caching controls and diagnostics NEW

90

Adds a unified `cache` setting and `Caching` capability for cross-provider prompt caching control, including an `openai_cache_instructions` parameter to place a prompt cache breakpoint on the instructions. Records prompt cache diagnostics in `provider_details` (on by default for OpenAI Responses, opt-in via `anthropic_cache_diagnostics`) and reports prompt-cache health per conversation via `pydantic_ai.cache.*` span attributes.

Enable cross-provider prompt caching on an Anthropic agent to reduce token costs on repeated instructions and tool definitions.

python

```python
from pydantic_ai import Agent
from pydantic_ai.capabilities import Caching

agent = Agent(
    'anthropic:claude-haiku-5-5',
    capabilities=[Caching(cache=True)],
)
```

— Names every setting and span attribute with a working example v2.55.0

02 Conversation persistence with Postgres backends NEW

80

Adds a `Conversation` class to carry and store a conversation across multiple runs, accepted via the `conversation=` parameter on every agent entry point, plus `PostgresStepStore` and `PostgresMediaStore` as hosted Postgres backends for persisting messages and media.

Persist a multi-turn conversation across separate agent runs so context carries over without re-sending full history manually.

```python
from pydantic_ai import Agent
from pydantic_ai.conversation import Conversation

agent = Agent('openai:gpt-6-sol')
conv = Conversation()

result1 = agent.run_sync('What is the capital of France?',
conversation=conv)
result2 = agent.run_sync('And what is its population?',
conversation=conv)
```

— Names class, param and both store backends with a runnable example v2.55.0

03 **WebRTC call controls for realtime and GPT-Live sessions** NEW

80

Adds `openai_live_idle_audio` so text-only GPT-Live sessions work without a microphone, a hang_up() method to end a WebRTC call from the server on gpt-realtime and GPT-Live, and seeds `message_history` into a WebRTC call through `answer_webrtc_offer`.

— Names all three surfaces but no usage example given v2.55.0

04 **FallbackModel attempt tracking in usage accounting** BREAKING

65

Each failed `FallbackModel` attempt is now recorded with its model, timing, error, and usage; rejected responses are counted in `RunUsage`, changing usage accounting.

— Names classes and fields affected by the accounting change v2.55.0

05 **Python 3.10 support dropped** BREAKING

60

Python 3.10 is no longer supported; all packages now require Python 3.11 or newer, and Python 3.10 installs will resolve to v2.54.0 or earlier.

— Clear version requirement and fallback behavior stated v2.55.0

06 **Tool requests moved under durable execution** BREAKING

60

Tool requests for `ExaSearch`, `YouSearch`, `LocalStack`, and other harness capabilities are now recorded under durable execution, changing replay behavior on recovery.

— Names affected integrations but no migration step given v2.55.0

THINNER COVERAGE BELOW

07 **Retained realtime audio bounded by max seconds** BREAKING

55

Retained realtime audio is now bounded by `retain_audio_max_seconds`, matching the behavior of retained images.

– Names the config key but no default value given v2.55.0

08 **LogfireMCP tool names prefixed** BREAKING

55

`LogfireMCP` tool names are now prefixed with `logfire_`, which will break any code referencing those tool names by their old unprefixed names.

– Exact prefix and breakage condition named v2.55.0

09 **FileUrl.media_type serialization change** BREAKING

55

`FileUrl.media_type` is now serialized as `null` for extensionless URLs instead of raising an exception.

– Names exact field and before/after behavior v2.55.0

10 **Realtime values sourced from session core** BREAKING

50

Realtime history, usage, and reply waits are now taken from the session core on OpenAI, Azure OpenAI, and xAI, changing how these values are sourced.

– Names providers but no concrete remediation v2.55.0

11 **Stable run and conversation IDs on re-execution** BREAKING

50

Default `run_id` and `conversation_id` are now kept stable when a durable run is re-executed, changing the identity of re-executed runs.

– Names fields but no concrete impact example v2.55.0

12 **OpenAIDecisionsModel backend** NEW

35

Adds `OpenAIDecisionsModel` backend with image input support.

– Bare name, no usage or config shown v2.55.0

└──▷ BREAKING ON UPGRADE

! Python 3.10 is no longer supported; all packages now require Python 3.11 or newer (Python 3.10 installs will resolve to v2.54.0 and earlier).

! Tool requests for `ExaSearch`, `YouSearch`, `LocalStack`, and other harness capabilities are now recorded under durable execution, changing replay behavior on recovery.

! Retained realtime audio is now bounded by `retain_audio_max_seconds`, matching the behavior of retained images.

! `LogfireMCP` tool names are now prefixed with `logfire_`, which will break any code referencing those tool names by their old unprefixed names.

! `FileUrl.media_type` is now serialized as `null` for extensionless URLs instead of raising an exception.

! Realtime history, usage, and reply waits are now taken from the session core on OpenAI, Azure OpenAI, and xAI, changing how these values are sourced.

! Default `run_id` and `conversation_id` are now kept stable when a durable run is re-executed, changing the identity of re-executed runs.

! Each failed `FallbackModel` attempt is now recorded with its model, timing, error, and usage; rejected responses are counted in `RunUsage`, changing usage accounting.

WAS THIS USEFUL? 👍 👎

assistant-ui

Feedback ⓘ `@assistant-ui/react@0.15.26` NOTES ↗

RANK

Assistant-ui provides React components and state management for chat interfaces with streaming, tool calls, and attachments.

> Assistant-ui marked its experimental APIs as deprecated to improve editor and linting feedback for developers.

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

01 **Experimental APIs marked deprecated** DEPRECATED 55

Every experimental API now carries `@deprecated Experimental since <date>`, so editors render it with a strikethrough instead of a removal warning, and `@typescript-eslint/no-deprecated` reports these APIs as deprecated.

— Names the exact JSDoc tag and lint rule involved `@assistant-ui/react@0.15.26`

WAS THIS USEFUL? 👍 👎

Cloudflare Agents

Feedback ⓘ agents@0.28.0 NOTES ↗ RANK

The Cloudflare Agents SDK builds stateful agents on Workers and Durable Objects, with persistent state, scheduling, WebSockets, and MCP server support.

> Cloudflare Agents 0.28.0 adds two new agent tools — a persistent browser session tool and a URL-to-text/markdown web fetch tool — and promotes several core modules to stable.

└──▷ **WHAT SHIPPED · 3 FEATURES** most completely described first ? what's the number?

01 web_fetch tool for URL retrieval and conversion NEW

86

New `agents/webfetch` package exposes a `web_fetch` tool that reads a URL and returns it as Markdown, JSON, or plain text, with HTML, PDF, and Office document conversion handled via Worker AI. Available via `agents/webfetch/pi`, `agents/webfetch/ai-sdk`, and `agents/webfetch/tanstack-ai`.

Let an agent fetch a live URL and receive its content as Markdown, JSON, or plain text — including PDFs and Office documents converted via Worker AI.

typescript

```typescript
import { web_fetch } from 'agents/webfetch/pi';

// Inside your agent definition, register the tool
const tools = [
  web_fetch()
];
```

— Names formats, conversion mechanism, and import paths with a code example.

agents@0.28.0

02 Persistent browser session tool for agents NEW

78

A new browserTool() lets an agent create and manipulate a persistent browser session, useful for login flows, scraping dynamic pages, or multi-step web automation. It's exposed via `agents/browser/pi`, with equivalents at `agents/browser/ai-sdk` and `agents/browser/tanstack-ai`.

Give an agent the ability to drive a persistent browser session — useful for login flows, scraping dynamic pages, or multi-step web automation.

```typescript
import { browserTool } from 'agents/browser/pi';

// Inside your agent definition, register the tool
const tools = [
  browserTool()
];
```

— Names exact import paths and shows a runnable snippet. `agents@0.28.0`

THINNER COVERAGE BELOW

03 **Core modules marked stable** IMPROVED 35

`agents/lifecycle`, `Scheduler` (`agents/schedules`), `State` (`agents/state`), `WebSockets` (`agents/websockets`), and `MCPClientManager` (`agents/mcp/client`) are now marked stable.

— Names the stabilized modules but gives no behavioural detail. `agents@0.28.0`

WAS THIS USEFUL?  

DEPLOY

◆ Agentic Infrastructure

HeyGen HyperFrames

Feedback ⓘ v0.8.145 NOTES ↗ RANK

HyperFrames is an open-source HTML-to-video renderer that runs in AI-agent workflows.

> HyperFrames shipped GPU-accelerated export for Studio and expanded its transcribe command with subtitle file output and speech-model visibility.

└──▷ WHAT SHIPPED · 2 FEATURES most completely described first ? what's the number?

01 GPU-accelerated export in Studio NEW

73

Studio export can now use GPU-accelerated encoding, equivalent to the `render --gpu` CLI flag, to speed up encoding of long compositions by offloading it from the CPU to the GPU.

Encode a long composition faster by offloading encoding to the GPU instead of the CPU.

```
$ npx hyperframes render --gpu my-composition.html
```

— Names exact flag and runnable command; mechanism is brief v0.8.145

02 Subtitle export and speech-model listing in transcribe NEW

72

The `transcribe` command gains a `--to` flag that writes subtitle files directly from a media file in `srt` or `vtt` format, and `models list` now shows which speech models `transcribe` can use so a model can be checked before a job is run.

Generate a ready-to-ship VTT subtitle file from a video in one step, skipping manual format conversion.

```
$ hyperframes transcribe interview.mp4 --to vtt
```

Check which speech models are available before choosing one for a transcription job.

```
$ hyperframes models list
```

— Names flag, formats, and command; limited detail on model differences v0.8.144

WAS THIS USEFUL? 👍 👎

OpenAI Python SDK

Feedback ⓘ v3.28.0 NOTES ↗ RANK

The OpenAI Python SDK provides typed Python clients for OpenAI model APIs, streaming, and asynchronous requests.

> OpenAI Python SDK added API client support for agent environment lifecycle management, including suspension, expiration, and prewarmed hosted environments.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Agent environment suspension and expiration** NEW 30

The API client now supports suspending and expiring agent environments.

— Names capability but no method, endpoint, or usage detail v3.28.0

02 **Prewarmed hosted environments support** NEW 30

Adds support for prewarmed hosted environments through the API.

— Bare feature name without mechanism or API specifics v3.27.0

WAS THIS USEFUL? 👍 👎

OpenAI TypeScript SDK

Feedback ⓘ v7.32.0 NOTES ↗

RANK

The OpenAI TypeScript SDK provides typed JavaScript and TypeScript clients for OpenAI APIs and streaming responses.

> The OpenAI TypeScript SDK v7.32.0 adds lifecycle controls for agent environments, including suspension and expiration.

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

01 **Agent environment suspension and expiration** NEW 30

Adds agent environment suspension and expiration to the API, enabling lifecycle control over agent environments.

— No endpoint, flag, or field names given. v7.32.0

WAS THIS USEFUL?  

Anthropic TypeScript SDK

Feedback ⓘ sdk-v0.133.0

NOTES ↗ RANK

The Anthropic TypeScript SDK provides typed JavaScript and TypeScript clients for Claude APIs, streaming, and tool use.

> Anthropic TypeScript SDK v0.133.0 adds typed support for unified analytics metrics and expands Managed Agents with workflow, multiagent, and thread-status capabilities.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Managed Agents workflow, multiagent and thread-status support**

NEW 30

Managed Agents gains support for workflows, multiagent configuration, and thread status filtering.

— Describes new capabilities but no concrete API or config names given sdk-v0.133.0

02 **Unified analytics types for Chat and Cowork** NEW 25

The SDK adds TypeScript types for Chat and Cowork unified analytics metrics.

— Names the metric area but no specific type names or fields sdk-v0.133.0

WAS THIS USEFUL? 👍 👎

Anthropic Python SDK

Feedback ⓘ v1.13.0 NOTES ↗ RANK

The Anthropic Python SDK provides typed Python clients for Claude APIs, streaming responses, and tool use.

> Anthropic Python SDK v1.13.0 adds typed support for unified analytics metrics and expands the Managed Agents API with workflow and multiagent configuration options.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 **Workflows and thread filtering in Managed Agents API** NEW

40

Adds `workflows`, multiagent configuration, and thread status filtering to the Managed Agents API.

— Names concrete additions but no mechanism or example v1.13.0

02 **Unified analytics metric types** NEW

35

Adds types for the Chat and Cowork unified analytics metrics to the API.

— Names two metric types but no usage detail v1.13.0

WAS THIS USEFUL? 👍 👎

◆ Inference Serving & Optimization

NVIDIA Dynamo

Feedback ⓘ v1.6.0-minimax-m3-dev.1 NOTES ↗

Dynamo is a datacenter-scale distributed inference serving framework with disaggregated prefill and decode, KV-cache-aware routing, and multi-engine support.

Adds `recipes/minimax-m3` Kubernetes recipes (`agg-gb200-agentic`, `disagg-gb200-agentic`) for GB200; improves prefill load accounting to exclude already-offloaded KV tokens.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch v1.6.0-minimax-m3-dev.1 https://github.com/ai-dynamo/dynamo.git
# already have the repo? check out this version:
$ git checkout v1.6.0-minimax-m3-dev.1
```

NEW SUPPORTED PLATFORM Adds `recipes/minimax-m3` Kubernetes recipes with `agg-gb200-agentic` (3× TP4, 400 GB CPU KV offload per replica, 12 GPUs) and `disagg-gb200-agentic` (3 prefill + 3 decode TP4, NIXL KV transfer, 24 GPUs) profiles for serving `nvidia/MiniMax-M3-NVFP4` on GB200.

Prefill load accounting now charges only tokens that still require GPU compute, excluding KV already resident on host or disk, improving routing accuracy for KV-offload workloads.

WAS THIS USEFUL? 👍 👎

FlashInfer

Feedback ⓘ v0.7.1 NOTES ↗ RANK

FlashInfer provides GPU kernels for language-model inference, including attention, sampling, and quantized operations.

> FlashInfer shipped a major expansion of DeepSeek-V4 sparse MLA attention — NVFP4 KV-cache quantization, a dedicated dsv4 decode kernel, and shared prefill/decode entry points — plus new SM120/SM121 NVFP4 and FP8 attention kernels, a Cake FMHA backend with request-ordered decode planning, a cuDNN-backed linear-attention prefill engine, and quantized expert-parallel MoE support for Rubin and Blackwell, alongside several breaking changes to MoE and MQA-logits APIs.

└──▷ **WHAT SHIPPED · 18 FEATURES** most completely described first ? what's the number?

01 **Sparse MLA decode for DeepSeek V4 (dsv4)** NEW 93

flashinfer.mla.trtllm_batch_decode_sparse_mla_dsv4() runs batched sparse MLA decode for DeepSeek V4, dispatching to TRTLLM-GEN kernels on SM100/SM103 and a packed sparse backend on SM120/SM121. `backend` accepts `'auto'`, `'trtllm-gen'`, `'cute-dsl'`, `'sparse'`, and `'cake'`; `kv_cache_format` accepts `'fp8'`, `'nvfp4'`, `'fp8_dsv41'`, and `'fp8_dsv41_fp4_ca'` on SM120/SM121. The `backend='cake'` path (`flashinfer.mla.cake_dsv4`) supports dense `[B, Q, H, D]` and ragged `[sum_q, H, D]` query layouts plus partial metadata tables. The `backend='cute-dsl'` (HCA) path adds `hca_swa_indices`, `hca_compressed_block_tables`, `hca_seq_lens`, `hca_is_causal`, `hca_use_persistent` (persistent tile scheduler, removing the `B * Q <= launch-grid` restriction), and `hca_sparse_indices_format='compressed-page-aligned'`. `dsv4_inv_rope_cos_sin_cache` triggers the TRTLLM-GEN RopeQuant epilogue (group-major FP8 E4M3 output with packed UE8M0 scales instead of BF16), and `enable_pdl` enables Programmatic Dependent Launch on the TRTLLM-GEN path.

Use the HCA (cute-dsl) backend on SM100/SM103 with the persistent scheduler to lift the B*Q grid-size limit and convert legacy page-aligned compressed metadata automatically.

```python
import flashinfer.mla

out = flashinfer.mla.trtllm_batch_decode_sparse_mla_dsv4(
    query=query,                          # [B, Q, num_heads,
512] BF16 or FP8 E4M3
    swa_kv_cache=swa_kv_cache,
    compressed_kv_cache=compressed_kv_cache,
    workspace_buffer=workspace_buffer,
    sparse_topk_lens=sparse_topk_lens,    # [B*Q] including 128
SWA slots
    seq_lens=seq_lens,                    # [B] INT32
    hca_swa_indices=hca_swa_indices,      # [B*Q, 128] INT32
    hca_compressed_block_tables=hca_compressed_block_tables,
    hca_seq_lens=hca_seq_lens,            # [B] INT32
    hca_is_causal=True,
    hca_use_persistent=True,
    hca_sparse_indices_format='compressed-page-aligned',
    bmm1_scale=1.0,
    bmm2_scale=1.0,
    backend='cute-dsl',
)
```

Decode on SM120/SM121 with NVFP4 KV cache format and a separate compressed segment, for bandwidth-efficient inference on Blackwell consumer GPUs.

```python
import flashinfer.mla

out = flashinfer.mla.trtllm_batch_decode_sparse_mla_dsv4(
    query=query,                          # [sum_q, num_heads,
512] BF16
    swa_kv_cache=swa_kv_cache,            # packed uint8, last
dim 384 (NVFP4)
    compressed_kv_cache=compressed_kv_cache,
    workspace_buffer=workspace_buffer,
    sparse_indices=sparse_indices,        # SWA segment indices
    swa_topk_lens=swa_topk_lens,          # [sum_q] INT32
    extra_sparse_indices=extra_sparse_indices,
    extra_sparse_topk_lens=extra_sparse_topk_lens,
    bmm1_scale=bmm1_scale,
    bmm2_scale=bmm2_scale,
    kv_cache_format='nvfp4',
    backend='sparse',
)
```

— Many named flags and two runnable cross-backend examples. product docs

02 Sparse and variable-length MLA decode controls NEW

93

`flashinfer.mla.trtllm_batch_decode_with_kv_cache_mla` gains `sparse_mla_top_k` for sparse MLA attention and `skip_softmax_threshold_scale_factor` skip-softmax sparsity (threshold = scale factor / context length). LSE output is available via `return_lse=True` plus a pre-allocated `lse` buffer across `cake`, `trtllm-gen`, `cute-dsl`, and `sparse` backends, with `return_lse_base` set to `'basee'` or `'base2'`. Variable-length query batches use `cum_seq_lens_q` (`[batch_size + 1]`, int32) and `max_q_len` on Blackwell, TRTLLM-GEN, and monolithic CuTeDSL backends. `cute_dsl_impl` selects `'auto'`, `'modular'`, or `'monolithic'`; `kv_scale_format` selects `'auto'`, `'pow2_fp32'`, or `'arbitrary_fp32'` for the SM120/SM121 packed v32/GLM sparse backend. A caller-owned `multi_ctas_kv_counter_buffer` (sized via `get_trtllm_gen_multi_ctas_kv_counter_bytes`) supports the `trtllm-gen` backend, and `use_fp16_softmax` selects the `Fp16Softmax` cubin variant for head dims `576/512` or `320/256`.

Run dense MLA decode on SM100/SM103 (Blackwell) and get back attention output plus LSE in base-2 units for use in speculative or multi-step pipelines.

python

```python
import torch
from flashinfer.mla import trtllm_batch_decode_with_kv_cache_mla

out, lse = trtllm_batch_decode_with_kv_cache_mla(
    query=query,                    # [batch, 1, num_heads,
kv_lora_rank + qk_rope_head_dim]
    kv_cache=kv_cache,              # [num_pages, page_size,
kv_lora_rank + qk_rope_head_dim]
    workspace_buffer=workspace,
    qk_nope_head_dim=128,
    kv_lora_rank=512,
    qk_rope_head_dim=64,
    block_tables=block_tables,
    seq_lens=seq_lens,
    max_seq_len=4096,
    sparse_mla_top_k=0,
    bmm1_scale=1.0,
    bmm2_scale=1.0,
    return_lse=True,
    return_lse_base='base2',
    backend='auto',
)
```

Run sparse MLA decode on SM100/SM103 to reduce KV read bandwidth at inference time by attending only to the top-k most relevant pages.

```python
import torch
from flashinfer.mla import trtllm_batch_decode_with_kv_cache_mla

out = trtllm_batch_decode_with_kv_cache_mla(
    query=query,                  # [batch, 1, num_heads,
kv_lora_rank + qk_rope_head_dim]
    kv_cache=kv_cache,
    workspace_buffer=workspace,
    qk_nope_head_dim=128,
    kv_lora_rank=512,
    qk_rope_head_dim=64,
    block_tables=sparse_block_tables,  # sparse page table
    seq_lens=seq_lens,
    max_seq_len=4096,
    sparse_mla_top_k=32,          # enables TRTLLM-GEN sparse
path on SM100/SM103
    bmm1_scale=1.0,
    bmm2_scale=1.0,
    backend='auto',
)
```

Use variable-length query batches (e.g. speculative decoding with multiple draft tokens per request) by supplying `cum_seq_lens_q` and `max_q_len` to skip host-side validation overhead.

```python
import torch
from flashinfer.mla import trtllm_batch_decode_with_kv_cache_mla

# cum_seq_lens_q: [0, q0, q0+q1, ...], shape [batch_size+1],
dtype int32
cum_seq_lens_q = torch.tensor([0, 3, 5, 9], dtype=torch.int32,
device='cuda')

out = trtllm_batch_decode_with_kv_cache_mla(
    query=query,                  # [total_q=9, num_heads,
head_dim_qk]
    kv_cache=kv_cache,
    workspace_buffer=workspace,
    qk_nope_head_dim=128,
    kv_lora_rank=512,
    qk_rope_head_dim=64,
    block_tables=block_tables,
    seq_lens=seq_lens,
    max_seq_len=4096,
    sparse_mla_top_k=0,
    bmm1_scale=1.0,
    bmm2_scale=1.0,
    cum_seq_lens_q=cum_seq_lens_q,
    max_q_len=4,                  # avoids host-side validation
of cum_seq_lens_q
    backend='trtllm-gen',
)
```

— Dense parameter list with three runnable decode examples. product docs

03 **Cake FMHA backend for batch decode and context prefill** NEW

93

`flashinfer.prefill.trtllm_batch_context_with_kv_cache` gains a `backend` parameter (default `'trtllm-gen'`) accepting `'trtllm-gen'` or `'cake'` for prefill. `trtllm_batch_decode_with_kv_cache` adds the same `backend="cake"` option for batch decode on SM103, with FP8 E4M3 KV cache support (page size 64, dedicated V/output scale, specialized JIT cache keys for split1-4 and short-shard retention) alongside BF16 Q/O. Request-ordered decode is planned via `flashinfer.CakeFmhaRequestOrderedDecodePlan` (built from `module_name`, `batch_size`, `q_len`, `workspace_parts`, `grid`, `total_tiles`, `write_lse`) and flashinfer.plan_cake_fmha_request_ordered_paged_decode() (`request_order_case` of `'identity'` or `'length_desc'`); the plan is passed to `trtllm_batch_decode_with_kv_cache` via `request_order` (int32 CUDA tensor mapping launch slot to logical request index, updatable in place between CUDA Graph

replays) and `request_order_plan`, supporting head dimension 256, page size 64, 8 query heads, 1 KV head, and `num_splits` of 1 or 6; omitting `request_order_plan` falls back to a safe single-split route, and LSE output is FP32 base-2.

Switch to the Cake FMHA kernel for prefill when benchmarking alternative attention backends.

```python
output = flashinfer.prefill.trtllm_batch_context_with_kv_cache(
    ...,
    backend='cake'
)
```

Build a graph-safe decode launch plan for a batch before dispatching FMHA kernels, controlling tiling and LSE output.

```python
from flashinfer import CakeFmhaRequestOrderedDecodePlan

plan = CakeFmhaRequestOrderedDecodePlan(
    module_name="my_fmha_module",
    batch_size=32,
    q_len=1,
    workspace_parts=4,
    grid=(128, 1, 1),
    total_tiles=512,
    write_lse=True,
)
```

Plan a CAKE FMHA decode schedule before CUDA Graph capture, ordering requests by descending sequence length.

```python
import flashinfer

kv_lens = [128, 64, 32, 16]
plan = flashinfer.plan_cake_fmha_request_ordered_paged_decode(
    kv_lens=kv_lens,
    q_len=1,
    request_order_case='length_desc',
    real_batch_size=len(kv_lens),
    write_lse=False,
)
# Capture CUDA Graph here; between replays, update request_order
tensor in place
```

Explicitly invoke the Cake FMHA backend for BF16 batch decode, bypassing automatic backend selection.

```python
output = trtllm_batch_decode_with_kv_cache(
    q, kv_cache,
    backend="cake"
)
```

Use request-ordered decode with a pre-built plan under CUDA Graph for optimized multi-split routing.

```python
plan = flashinfer.plan_cake_fmha_request_ordered_paged_decode(
    ...  # fixed tensor/layout bindings
)
# One eager invocation to initialize TMA descriptors before graph capture
output = trtllm_batch_decode_with_kv_cache(
    q, kv_cache,
    backend="cake",
    request_order=request_order_tensor,
    request_order_plan=plan
)
# Subsequent CUDA Graph replays may update request_order contents in place
```

— All classes, functions and params named with five runnable examples. product docs

04 **TRT-LLM FMHAv2 prefill with paged KV and sparse attention**

NEW 92

flashinfer.prefill.trtllm_fmha_v2_prefill() runs TRT-LLM FMHAv2 prefill attention, accepting `input_layout` values `PACKED_QKV`, `CONTIGUOUS_Q_KV`, `Q_PAGED_KV_HND`, `Q_PAGED_KV_NHD`, and `SEPARATE_Q_K_V`. Paged KV cache is supported via `block_tables` with a stacked-pool 5D `paged_KV` tensor (`[batch_size, max_num_pages_per_seq]`) or separate (k_cache, v_cache) layouts. `mask_mode` accepts `causal`, `sliding_window` (via `window_left`), and `chunked` (via `chunked_attention_size`, power-of-2, SM90/Hopper only). `save_softmax_stats=True` returns (attention_output, softmax_stats_LSE); `skip_softmax_threshold_scale_factor` skips softmax/BMM2 when `exp(local_max - global_max) < skip_softmax_threshold_scale_factor / seqlen`; and `pos_encoding_mode='alibi'` plus `logits_soft_cap_scale` add ALiBi encoding and logits soft-capping.

Run TRT-LLM FMHAv2 prefill with paged KV cache and retrieve softmax LSE stats for speculative decoding or verification passes.

```python
import torch
import flashinfer.prefill

output, lse = flashinfer.prefill.trtllm_fmha_v2_prefill(
    qkv=(q, paged_kv),
    input_layout='Q_PAGED_KV_NHD',
    workspace_buffer=workspace,
    seq_lens=seq_lens,
    max_q_len=max_q_len,
    max_kv_len=max_kv_len,
    bmm1_scale=1.0 / (head_dim ** 0.5),
    bmm2_scale=1.0,
    batch_size=batch_size,
    cum_seq_lens_q=cum_seq_lens_q,
    cum_seq_lens_kv=cum_seq_lens_kv,
    block_tables=block_tables,
    save_softmax_stats=True,
)
```

Use sparse attention to skip softmax/BMM2 for tokens far below the global max, reducing compute on long-context prefill.

```python
import flashinfer.prefill

output = flashinfer.prefill.trtllm_fmha_v2_prefill(
    qkv=(q, k, v),
    input_layout='SEPARATE_Q_K_V',
    workspace_buffer=workspace,
    seq_lens=seq_lens,
    max_q_len=max_q_len,
    max_kv_len=max_kv_len,
    bmm1_scale=1.0 / (head_dim ** 0.5),
    bmm2_scale=1.0,
    batch_size=batch_size,
    cum_seq_lens_q=cum_seq_lens_q,
    cum_seq_lens_kv=cum_seq_lens_kv,
    skip_softmax_threshold_scale_factor=8.0,
)
```

Enable sliding-window causal attention for long-context inference where only recent tokens are attended to.

python

```python
import flashinfer.prefill

output = flashinfer.prefill.trtllm_fmha_v2_prefill(
    qkv=qkv,
    input_layout='PACKED_QKV',
    workspace_buffer=workspace,
    seq_lens=seq_lens,
    max_q_len=max_q_len,
    max_kv_len=max_kv_len,
    bmm1_scale=1.0 / (head_dim ** 0.5),
    bmm2_scale=1.0,
    batch_size=batch_size,
    cum_seq_lens_q=cum_seq_lens_q,
    cum_seq_lens_kv=cum_seq_lens_kv,
    mask_mode='sliding_window',
    window_left=4096,
)
```

— Every layout, mask mode and sparsity flag named with three examples. product docs

05 **AlphaMoE NVFP4 fused MoE traces** NEW 86

Adds `alphamoe_nvfp4_aligned_moe` and `alphamoe_nvfp4_routed_moe` traces for AlphaMoE NVFP4 quantized MoE at shapes `M=8` ,128,512 / `N=1024` / `K=6144` / `E=256` / `top_k=8` / `block_m=8.` Optional `w1_scale_prepared` and `w2_scale_prepared` uint8 fields retain raw scale inputs and physical dimensions on both traces. Routed traces add `w1_data_prepared` (`[E*(N/128)*(K/256),128,128]` , from `prepare_nvfp4_w1_data`), and for the 512-token routed shape, `w1_gate_up_data_prepared` (`[E*(N/256)*(K/256),256,128]` , from `prepare_nvfp4_w1_gate_up_data`) and `w1_gate_up_scale_prepared` (`[E*(N/256)*(K/256),32,128]` , from `prepare_nvfp4_w1_gate_up_scales`).

Prepare NVF4 W1 data and scale buffers once after weight load, then replay routed AlphaMoE traces for the 512-token shape without re-deriving on every request.

```python
w1_data_prepared = prepare_nvfp4_w1_data(w1_packed_raw)
w1_gate_up_data_prepared =
prepare_nvfp4_w1_gate_up_data(w1_scales_prepared,
raw_weight_shape)
w1_gate_up_scale_prepared =
prepare_nvfp4_w1_gate_up_scales(w1_scales_prepared,
raw_weight_shape)
```

— Exact trace names, shapes and prep functions given. product docs

06 **MLA prefill entry point with distributed context parallelism**

NEW 85

`trtllm_prefill_with_kv_cache_mla` (exposed through both `flashinfer.mla` and `flashinfer.prefill`) shares the MLA decode API signature and supports incremental prefill and MTP. It exposes `sparse_mla_top_k` and `sparse_mla_top_k_lens` for sparse top-k attention; `enable_dcp` , `cp_world` , `cp_rank` , and `causal_seqlens_kv_global` for distributed context parallelism; `skip_softmax_threshold_scale_factor` , `use_fp16_softmax` , `bmm1_scale` / `bmm2_scale` for numerics control; and `cute_dsl_impl` / `kv_scale_format` (both default `'auto'`). With `backend='auto'` , multi-token prefill (`q_len_per_request > 1`) requires an explicit compatible backend since XQA only supports `q_len_per_request == 1` .

Run sparse MLA prefill with top-k attention to reduce KV read cost during long-context incremental prefill.

```python
import torch
import flashinfer.mla

out = flashinfer.mla.trtllm_prefill_with_kv_cache_mla(
    query=query,
    kv_cache=kv_cache,
    workspace_buffer=workspace_buffer,
    qk_nope_head_dim=128,
    kv_lora_rank=512,
    qk_rope_head_dim=64,
    block_tables=block_tables,
    seq_lens=seq_lens,
    max_seq_len=4096,
    sparse_mla_top_k=64,
    backend='auto',
)
```

Enable distributed context parallelism across two ranks during MLA prefill for long-sequence tensor-parallel inference.

```python
import flashinfer.mla

out = flashinfer.mla.trtllm_prefill_with_kv_cache_mla(
    query=query,
    kv_cache=kv_cache,
    workspace_buffer=workspace_buffer,
    qk_nope_head_dim=128,
    kv_lora_rank=512,
    qk_rope_head_dim=64,
    block_tables=block_tables,
    seq_lens=seq_lens,
    max_seq_len=8192,
    sparse_mla_top_k=0,
    enable_dcp=True,
    cp_world=2,
    cp_rank=0,
    causal_seqlens_kv_global=causal_seqlens_kv_global,
)
```

— All parameters named with two runnable examples. v0.7.1

07 **FP8 self-attention kernel for SM120/SM121** NEW 83

`flashinfer.prefill.fmha_v2_prefill_sm120` runs FP8 E4M3 self-attention on SM120/SM121 with BF16 output. It supports persistent device-side FP32 scale tensors via `scale_bmm1_d` (fused q_scale * k_scale / sqrt(head_dim)) and `scale_bmm2_d` (V dequantization scale) as alternatives to host-encoded `scale_bmm1` / `scale_bmm2`, optional LSE output via `return_lse=True` with a caller-owned `lse` tensor of shape `[B, S, H, 2]` FP32, and `causal=False` for dense/bidirectional attention.

Run FP8 E4M3 causal self-attention on an SM120 GPU and retrieve both the output and log-sum-exp for downstream use (e.g. speculative decoding or attention composition).

python

```python
import torch
import flashinfer.prefill as fp

B, S, H, D = 2, 512, 16, 128
q = torch.randn(B, S, H, D, dtype=torch.float8_e4m3fn,
device='cuda')
k = torch.randn(B, S, H, D, dtype=torch.float8_e4m3fn,
device='cuda')
v = torch.randn(B, S, H, D, dtype=torch.float8_e4m3fn,
device='cuda')
out = torch.empty(B, S, H, D, dtype=torch.bfloat16,
device='cuda')
lse = torch.empty(B, S, H, 2, dtype=torch.float32, device='cuda')

result, lse_out = fp.fmha_v2_prefill_sm120(
    query=q, key=k, value=v, out=out,
    num_heads=H, head_dim=D, seq_len=S,
    scale_softmax=1.0 / (D ** 0.5),
    causal=True, return_lse=True, lse=lse
)
```

Use persistent device-side scale tensors (`scale_bmm1_d` , `scale_bmm2_d`) as model weights to avoid repeated host-to-device transfers during inference.

```python
import torch
import flashinfer.prefill as fp

B, S, H, D = 1, 256, 8, 64
q = torch.randn(B, S, H, D, dtype=torch.float8_e4m3fn,
device='cuda')
k = torch.randn(B, S, H, D, dtype=torch.float8_e4m3fn,
device='cuda')
v = torch.randn(B, S, H, D, dtype=torch.float8_e4m3fn,
device='cuda')
out = torch.empty(B, S, H, D, dtype=torch.bfloat16,
device='cuda')

# Fused q_scale * k_scale / sqrt(head_dim)
scale_bmm1_d = torch.tensor([0.00390625 * 0.00390625 / (D **
0.5)], dtype=torch.float32, device='cuda')
scale_bmm2_d = torch.tensor([0.00390625], dtype=torch.float32,
device='cuda')

result = fp.fmha_v2_prefill_sm120(
    query=q, key=k, value=v, out=out,
    num_heads=H, head_dim=D, seq_len=S,
    scale_softmax=1.0 / (D ** 0.5),
    causal=True,
    scale_bmm1_d=scale_bmm1_d,
    scale_bmm2_d=scale_bmm2_d
)
```

— Named scale tensors and output shapes with two runnable examples. product docs

08 **NVFP4 attention quantization and forward kernel on SM120** NEW

82

`flashinfer.nvfp4_attention_sm120.nvfp4_attention_sm120_quantize_qkv` preprocesses and quantizes Q/K/V for SM120 NVFP4 attention; its `per_block_mean` flag (default `True`) centers Q per 128-token block, while `False` centers Q once across the full sequence, changing the correction tensor shape from `[batch, num_qo_heads, M_pad, N_pad]` to `[batch, num_qo_heads, N_pad]`. `nvfp4_attention_sm120_fwd` then runs the attention, with optional `causal` masking, the same `per_block_mean` mode, `unpadded_k_len` to mask a logical K/V length shorter than the padded extent, and `return_lse` to also return the log-sum-exp tensor.

Run causal NVFP4 attention on SM120 hardware, retrieving only the attention output (no log-sum-exp) for inference.

```python
from flashinfer.nvfp4_attention_sm120 import
nvfp4_attention_sm120_fwd

out = nvfp4_attention_sm120_fwd(
    q_fp4=q_fp4,
    k_fp4=k_fp4,
    v_fp4_t=v_fp4_t,
    q_scale=q_scale,
    k_scale=k_scale,
    v_scale_t=v_scale_t,
    qk_correction=qk_correction,
    causal=True,
    out_dtype=torch.bfloat16,
    return_lse=False,
)
```

Run NVFP4 attention with a logical K/V length shorter than the padded extent, masking trailing pad tokens before softmax.

```python
from flashinfer.nvfp4_attention_sm120 import
nvfp4_attention_sm120_fwd

out, lse = nvfp4_attention_sm120_fwd(
    q_fp4=q_fp4,
    k_fp4=k_fp4,
    v_fp4_t=v_fp4_t,
    q_scale=q_scale,
    k_scale=k_scale,
    v_scale_t=v_scale_t,
    qk_correction=qk_correction,
    unpadded_k_len=actual_seq_len,
    return_lse=True,
)
```

Quantize Q/K/V into NVFP4 before running SM120 attention — use per-block centering (default) for long sequences where block-level statistics give better accuracy.

```python
import torch
from flashinfer.nvfp4_attention_sm120 import
nvfp4_attention_sm120_quantize_qkv

batch, num_qo_heads, num_kv_heads, seq_q, seq_kv, head_dim = 2,
16, 16, 512, 512, 128
q = torch.randn(batch, num_qo_heads, seq_q, head_dim,
dtype=torch.bfloat16, device='cuda')
k = torch.randn(batch, num_kv_heads, seq_kv, head_dim,
dtype=torch.bfloat16, device='cuda')
v = torch.randn(batch, num_kv_heads, seq_kv, head_dim,
dtype=torch.bfloat16, device='cuda')

q_fp4, k_fp4, v_fp4_t, q_scale, k_scale, v_scale_t, qk_correction
= \
    nvfp4_attention_sm120_quantize_qkv(q, k, v,
per_block_mean=True)
```

Use `per_block_mean=False` for short sequences where a single global centering is sufficient and you want the smaller `[batch, num_qo_heads, N_pad]` correction tensor.

```python
q_fp4, k_fp4, v_fp4_t, q_scale, k_scale, v_scale_t, qk_correction
= \
    nvfp4_attention_sm120_quantize_qkv(q, k, v,
per_block_mean=False)
# qk_correction.shape == (batch, num_qo_heads, N_pad)
```

— Two related functions and their flags named with four examples. product docs

09 **NVFP4 quantization for sparse MLA KV cache** NEW 81

flashinfer.mla.nvfp4_quantize_pack_sparse_mla_cache(latent_kv, kv_layout) quantizes DeepSeek-V4 latent-KV pages from BF16 to packed E2M1 with E4M3 scales (NVFP4), taking a `[num_pages, page_size, 512]` BF16 tensor and returning an opaque uint8 paged cache of shape `[num_pages, page_size, 384]` in `'HND'` or `'NHD'` layout.

Compress a freshly filled DeepSeek-V4 KV page buffer to NVFP4 before passing it to a sparse MLA attention kernel.

```python
import torch
import flashinfer.mla

# latent_kv: BF16, shape [num_pages, page_size, 512], on CUDA
latent_kv = torch.randn(32, 16, 512, dtype=torch.bfloat16,
device='cuda')

# Quantize to NVFP4 packed cache in HND layout
nvfp4_cache =
flashinfer.mla.nvfp4_quantize_pack_sparse_mla_cache(
    latent_kv,
    kv_layout='HND'
)
# nvfp4_cache: uint8, shape [32, 16, 384]
```

— Exact signature, shapes and dtypes with a runnable example. product docs

10 **cuDNN linear-attention prefill backend** NEW 80

recurrent_kda(backend="cudnn") and chunk_gated_delta_rule(backend="cudnn") run Kimi Delta Attention and Gated DeltaNet prefill on cuDNN's open-source fused linear-attention engine on Blackwell SM100, including CUDA 12 builds; `chunk_gated_delta_product` and `chunk_gated_delta_rule2` add Gated Delta Product and Gated Delta Rule 2 prefill on the same engine. Requires `nvidia-cudnn-frontend>=1.30.0`, installed with the `cutedsl` extra via `pip install -U 'nvidia-cudnn-frontend[cutedsl]>=1.30.0'`.

Run Kimi Delta Attention prefill on a Blackwell GPU using the cuDNN backend, which also works on CUDA 12 builds where the default kernel requires CUDA 13.

```python
import flashinfer

# Select cuDNN backend for Kimi Delta Attention prefill
out = flashinfer.recurrent_kda(q, k, v, backend="cudnn")

# Select cuDNN backend for Gated DeltaNet prefill
out2 = flashinfer.chunk_gated_delta_rule(q, k, v,
backend="cudnn")
```

— Named entry points, install requirement and runnable example. v0.7.1

11 **Quantized expert-parallel MoE on Rubin and Blackwell** NEW 72

`MoEEpLayer` adds NVFP4 and MXFP8 MegaMoE experts on Rubin SM107 with BF16 combine and output. `expert_map` shards experts across GPUs for W4A16 fused MoE on Blackwell SM120/SM121 multi-GPU workstations. New CuTe DSL NVFP4 W4A16 and W4A4 MoE backends on SM12x read a single prepared expert layout, letting a serving stack choose W4A16 for decode and W4A4 for prefill from one weight copy. CUDA Graph capture is now supported for the split MoE layer with both NCCL-EP and NIXL-EP backends.

Enable a per-model NVFP4 4over6 recipe so two models with different quantization configs can serve simultaneously in one process.

python

```python
from flashinfer.quantization import NVFP44Over6Config,
QuantConfig

# Attach an explicit recipe to a model's MoE config
quant_cfg = QuantConfig(nvfp4_recipe=NVFP44Over6Config(...))

# Pass it to the MoE layer — process-wide env vars apply only
when no recipe is given
moe_layer = MoEEpLayer(..., quant_config=quant_cfg)
```

— Names four MoE surfaces and GPU targets with one example. v0.7.1

12

BatchMLAPagedAttentionWrapper with automatic backend selection

NEW

68

`BatchMLAPagedAttentionWrapper` provides a plan()/run() lifecycle to prepare and reuse TRTLLM-GEN, XQA, and monolithic or modular CuTe DSL backends, with CUDA Graph capture support. plan() accepts `backend="auto"` on SM100 to automatically select among all available backends for the planned shapes and features.

Let BatchMLAPagedAttentionWrapper automatically pick the best MLA backend for your shapes on SM100, then reuse the plan across decode steps for CUDA Graph capture.

```python
import flashinfer

wrapper = flashinfer.mla.BatchMLAPagedAttentionWrapper(...)

# plan() auto-selects among TRTLLM-GEN, XQA, and CuTe DSL on
SM100
wrapper.plan(..., backend="auto")

# reuse the plan in run() — compatible with CUDA Graph capture
out = wrapper.run(q, kv_cache)
```

— Names the class and lifecycle methods with a short example. v0.7.1

13 **Env var to disable cuTile autotuning** NEW 62

`FLASHINFER_CUTILE_AUTOTUNE_DISABLED` disables per-process autotuning of tile shape, CTA count, and occupancy for `backend="cutile"` GEMM/BMM ops, forcing the built-in default config instead. Set via `export FLASHINFER_CUTILE_AUTOTUNE_DISABLED=1`.

Disable cuTile autotuning overhead during benchmarking to get deterministic, tuning-free timing for GEMM/BMM ops.

```
$ export FLASHINFER_CUTILE_AUTOTUNE_DISABLED=1
```

— Exact env var and shell command, but thin on mechanism. product docs

14 **Breaking changes to MoE wrapper and dispatch APIs** BREAKING 62

`cute_dsl_fused_moe_bf16` now takes `tactic` for tuning, and `CuteDslBf16MoEWrapper` reorders its constructor arguments — callers must pass `output_dtype`, `enable_pdl`, and `use_fused_finalize` by keyword. `moe_a2a_dispatch` now takes `recv_view_cache` by keyword only.

— Exact signature changes named, no migration example. v0.7.1

15 **MQA logits API signature changes** BREAKING 62

`fp8_paged_mqa_logits` and `fp4_paged_mqa_logits` must now be called with `block_tables`, `seq_lens`, and `max_seq_len`; positional callers must pass the page table before sequence lengths, and `num_epi_subtiles` is removed. FP4 callers must also rename `sf_q` to `q_sf`.

— Exact rename and ordering change named, no code sample. v0.7.1

16 **Fused Kimi K3 speculative decode kernel** NEW 61

`packed_fused_kda_decode` on Blackwell SM10x fuses short convolution, Kimi Delta Attention, and gated RMS normalization during speculative verification, supporting ragged batches, accepted-token cache updates, CUDA Graph capture, and FP32 or BF16 recurrent state.

— Named function and scope but no usage example given. v0.7.1

17 **Natural-log LSE option for cuDNN prefill** IMPROVED 60

`cudnn_batch_prefill_with_kv_cache` adds an `lse_base` parameter accepting `'log2'` (default, matches other FlashInfer backends) or `'ln'`, which returns cuDNN's native natural-log stats directly and skips the conversion kernel.

Skip the log2-conversion kernel on the cuDNN path to reduce overhead when your downstream code can consume natural-log LSE directly.

```python
output, lse = cudnn_batch_prefill_with_kv_cache(
    ...,
    lse_base='ln'
)
```

— Single named parameter and values with a short example. product docs

THINNER COVERAGE BELOW

18 **Removal of prims batch decode wrappers** BREAKING 50

`prims_ts_batch_decode_with_kv_cache` and `prims_ts_batch_mla_decode_with_kv_cache` are removed; callers must migrate to the corresponding paged decode wrappers.

— Names removed functions but no replacement signature given. v0.7.1

└──▷ BREAKING ON UPGRADE

- ! Install `nvidia-cudnn-frontend>=1.30.0`; the cuDNN linear-attention backend additionally requires the `cutedsl` extra: `pip install -U 'nvidia-cudnn-frontend[cutedsl]>=1.30.0'`.
- ! `cute_dsl_fused_moe_bf16` now takes `tactic` for tuning and `CuteDslBf16MoEWrapper` reorders its constructor arguments — pass `output_dtype`, `enable_pdl`, and `use_fused_finalize` by keyword.
- ! `moe_a2a_dispatch` now takes `recv_view_cache` by keyword only.
- ! Calls to `fp8_paged_mqa_logits` and `fp4_paged_mqa_logits` must use `block_tables`, `seq_lens`, and `max_seq_len`; pass the page table before sequence lengths in positional

calls; remove `num_epi_subtiles`. FP4 callers must also rename `sf_q` to `q_sf`.

! `prims_ts_batch_decode_with_kv_cache` and `prims_ts_batch_mla_decode_with_kv_cache` are removed — migrate to the corresponding paged decode wrappers.

WAS THIS USEFUL? 👍 👎

◆ Local LLM Runtimes

mlxtop

Feedback ⓘ v2.1.1 NOTES ↗ RANK

The mlxtop terminal UI monitors local LLMs on Apple Silicon, showing loaded models, memory use, GPU activity, and generation speed.

> mlxtop 2.1.1 adds a `mlxtop doctor` diagnostic subcommand, macOS compression monitoring with tunable thresholds, and a reorganized Overview with threshold-colored charts and a plain-language health verdict.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 **macOS compression monitoring and tuning** NEW 95

Adds a macOS-only compression chart to Overview showing compress and decompress traffic, stored vs compressor bytes, and ratio, plus `compression_warn_rate` and `compression_warn_exit` config keys in `~/.config/mlxtop/config.json` to tune when compression churn raises and clears the COMPRESSION ACTIVE finding (defaults: 64 MiB/s warn, 32 MiB/s exit); Linux shows memory and paging instead.

Lower the macOS compression-churn threshold so COMPRESSION ACTIVE fires earlier on machines with limited RAM, reducing the risk of undetected memory pressure.

```json
{
  "compression_warn_rate": 33554432,
  "compression_warn_exit": 16777216
}
```

— Names config keys, file path, defaults, and chart contents with a config example.

v2.1.1

02 **mlxtop doctor diagnostic subcommand** NEW 90

Adds a `d` keybind and `mlxtop doctor` subcommand that explains the current assessment, selected runtime and endpoint, authentication status, and setup steps; returns exit code 1 for invalid config, incomplete host counters, or a failed primary runtime connection.

Diagnose why a runtime metric is missing or a connection is failing without opening the TUI — useful in headless or CI environments.

```
$ MLXTOP_PROVIDER=ollama mlxtop doctor
```

— Named subcommand, keybind, exit codes, and a runnable command. v2.1.1

THINNER COVERAGE BELOW

03 **Threshold-colored charts across Overview** IMPROVED 55

Applies threshold colors to every chart in Overview — GPU, paging, compression, memory, and cache use configured bands; generation and prefill are graded yellow at −10% and red at −30% against their rolling baseline.

— Names thresholds and charts but no command or config to act on. v2.1.1

04 **Overview layout restructured around host resources** IMPROVED 55

Restructures Overview to lead with host resources (RAM composition: wired, app, compressed, cache, free; GPU wired limit; compression; paging) followed by a compact throughput row, with all panels sharing one aligned three-column grid.

— Describes new layout in detail but no navigation path or command. v2.1.1

05 **Human-readable assessment verdict** NEW 45

Adds a human-readable assessment verdict (Healthy · no bottleneck or named next check) that replaces itself with a critical alarm when thresholds are breached, without hiding SYSINFO.

— Describes behavior but no concrete command or config shown. v2.1.1

WAS THIS USEFUL? 👍 👎

◆ Inference Providers & GPU Clouds

Cloudflare Workers AI

Feedback ⓘ

Workers AI - Clef-omni adds audio and video input, Clef-flash is now cheaper, and Clef i

BLOG ↗

Cloudflare Workers AI runs open-weight models, including the Clef decision models, as serverless inference on Cloudflare's global network.

Clef-omni brings audio/video input to Workers AI decision models; Clef-flash is cheaper at $0.038/M tokens; Clef is up to 2x faster.

NEW INTEGRATION Adds `@cf/cloudflare/clef-omni`, accepting `audio` (WAV/MP3) and `videos` (MP4/WebM) fields in the System One API for single-request multi-modal decisions.

Cuts `@cf/cloudflare/clef-flash` price to $0.038/M input tokens (from $0.090); hosted context window reduced to 24K tokens.

Improves `@cf/cloudflare/clef` inference speed up to 2x via migration to SGLang, with no model weight changes.

└─▷ BREAKING ON UPGRADE

! The hosted `@cf/cloudflare/clef-flash` context window is reduced from 64K to 24K tokens; requests exceeding 24K input tokens will need to switch to `@cf/cloudflare/clef` or self-host.

WAS THIS USEFUL? 👍 👎

fal

Feedback ⓘ Creator and Viewer Team Roles NOTES ↗ RANK

Provides hosted generative-media model APIs and a serverless GPU platform for deploying custom inference workloads.

> fal expanded team access control with two new granular roles and overhauled API key permissioning with seven presets, custom policies, model-scoped limits, and expiry.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 Key permission presets and custom policies BREAKING

85

fal introduces seven key permission presets — `API`, `SERVERLESS`, `DEPLOY`, `COMPUTE`, `BILLING`, `READONLY`, and `FULL` — replacing the old `ADMIN` and `API` scopes when creating keys. Custom policy keys let you select exact permissions individually instead of a preset, keys can be scoped to specific models and set to expire (any model-limited key is automatically treated as a custom policy key with permissions frozen at creation time), and the Platform API reference now annotates each endpoint with the permissions it requires and which presets include them.

Create a least-privilege key for a CI/CD pipeline that only needs to call inference endpoints — no billing or deployment access.

📍 In the fal dashboard, go to Keys › Create Key, select the `SERVERLESS` preset, optionally restrict to specific models, and set an expiry date.

Lock down a contractor key so it can only call one specific model and automatically expires after their engagement ends.

📍 In the fal dashboard, go to Keys › Create Key, choose the `API` preset, enter the target model name under `Limit to specific models`, and set an expiration date — this creates a custom policy key with permissions frozen at creation.

— Names all preset values and old scopes replaced; UI steps given.

Key Presets and Custom Policies

02 Creator and Viewer team roles NEW

75

Admins can now assign two new team roles alongside existing ones: `Creator`, which grants access to Playground, Sandbox, and Workflows but blocks request logs, API key management, billing, members, and role changes; and `Viewer`, which gives read-only access to request logs and the library with no ability to run models or modify team resources.

Restrict a contractor to running models in Playground and Workflows without exposing billing or API keys.

> 📍 In the team dashboard, go to Members, select the member, open the Role dropdown, and choose `Creator`.

Give a stakeholder read-only visibility into request logs and the model library without allowing any model execution or changes.

> 📍 In the team dashboard, go to Members, select the member, open the Role dropdown, and choose `Viewer`.

— Clear role scopes named, UI path given but no API/config key.

`Creator and Viewer Team Roles`

WAS THIS USEFUL? 👍 👎

DATA

◆ Vector DBs & RAG

Unstructured

Feedback ⓘ 0.27.25 NOTES ↗ RANK

Unstructured is an open-source ETL library that converts PDFs, images, and office documents into chunked, structured elements for LLM ingestion.

Adds WEBP image partitioning support, treating WEBP files the same as PNG and JPEG in the partition pipeline.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 0.27.25 https://github.com/Unstructured-IO/unstructured.git
# already have the repo? check out this version:
$ git checkout 0.27.25
```

NEW SUPPORTED PLATFORM WEBP image partitioning

WAS THIS USEFUL? 👍 👎

Volcengine OpenViking

Feedback ⓘ v0.5.0 NOTES ↗ RANK

OpenViking is a context database that manages agent memory, knowledge retrieval, and skills.

> OpenViking v0.5.0 introduces a standalone Gateway process that injects memory into any OpenAI- or Anthropic-compatible client via a base-URL swap, alongside a string of breaking changes to Working Memory defaults, search filtering, MCP return values, and plugin history handling.

└──▷ **WHAT SHIPPED · 13 FEATURES** most completely described first ? what's the number?

01 Standalone Gateway process for memory injection NEW

95

OpenViking ships a standalone Gateway process that injects memory into any OpenAI- or Anthropic-compatible client via a base-URL swap, requiring no prompt or code changes. It is activated via the `OPENVIKING_GATEWAY_ENCRYPTION_KEY` (Fernet key) and `OPENVIKING_GATEWAY_ADMIN_TOKEN` (≥32 chars) environment variables plus `enabled: true` in the `gateway` section of `ov.conf`, and can be deployed via the new pip extras `gateway` and `gateway-fast`, a `gateway` Docker Compose profile, or the Helm chart (promoted to `0.2.0`) with `gateway.*` sidecar support. Studio gained a Gateway section (Overview, Upstreams, Profiles, Keys, Requests, Connect) for managing upstreams, context profiles and gateway keys, backed by a server-side admin proxy at `/api/v1/admin/gateway/*`.

Route an OpenAI-compatible client through the gateway to get automatic memory injection without changing any prompts or code.

```
$ pip install 'openviking[gateway]'
export OPENVIKING_GATEWAY_ENCRYPTION_KEY='<fernet-key>'
export OPENVIKING_GATEWAY_ADMIN_TOKEN='<32-char-token>'
openviking-gateway
```

Enable the gateway in `ov.conf` before starting it, supplying the required encryption and admin credentials.

```ini
[gateway]
enabled = true
```

— names env vars, config key, endpoint, deployment options and UI path v0.5.0

02 Per-commit Working Memory override NEW

90

`POST /api/v1/sessions/{session_id}/commit` gains an `enable_working_memory` field (`true` / `false` / `null`) to override Working Memory for a single commit; the response now includes `effective_enable_working_memory` .

Trigger Working Memory extraction for a single commit on a session that otherwise has Working Memory turned off.

```
$ curl -X POST "$OV_URL/api/v1/sessions/$SID/commit" \
  -H "X-API-Key: $KEY" \
  -H 'Content-Type: application/json' \
  -d '{"enable_working_memory": true}'
```

— named endpoint, field and response field with runnable curl example v0.5.0

03 **Context-mode search excludes system-preset directories**

78

BREAKING

`/api/v1/search/search` with `mode: "context"` no longer returns system-preset directories (`viking://resources` , `viking://agent` , user root and its first-level directories, and their subtrees); unused per-category quota is now redistributed to the best results from other categories. Results from this endpoint may differ from v0.4.23.

— names endpoint, mode and excluded paths with behavior change v0.5.0

04 **Plugin integrations default to host-managed history** BREAKING

73

Claude Code, Codex, DSH, OpenCode, and pi plugin commits now send `keep_recent_count: 0` , and the `commitKeepRecentCount` setting and `OPENVIKING_COMMIT_KEEP_RECENT_COUNT` environment variable are no longer read. These integrations (pi in non-takeover mode) now default to letting the host manage history and compaction rather than OpenViking, though existing explicit config and environment variables still take precedence.

— names affected plugins, config key, env var and default change v0.5.0

05 **Local provider subcommands for private deployments** NEW

70

New `hermes openviking local status/start/stop/restart` subcommands run OpenViking and a local BGE embedding model via the new Quick Local external provider, for use in private environments.

— names subcommands and provider but no example given v0.5.0

06 **Working Memory disabled by default** BREAKING 70

`memory_policy.working_memory.enabled` default changed from `true` to `false`; deployments that need Working Memory must re-enable it explicitly. Policies previously serialized with `working_memory` omitted (because it defaulted to `true`) will now be read as `false` after upgrade.

— names exact config key and before/after default with migration note v0.5.0

07 **Tool-loop limits for gateway's internal tool calls** NEW 68

New context-profile fields `tool_max_rounds`, `tool_total_seconds`, and `tool_total_tokens` (default empty = no limit) cap the gateway's internal OpenViking tool loop, alongside the existing per-call `tool_timeout_seconds` (30s) timeout.

— names all four fields and defaults, no example v0.5.0

08 **MCP remember returns task_id instead of fixed string** BREAKING 65

MCP `remember` no longer returns the fixed string `Stored N message(s) and committed for memory extraction.`; it now returns a submission confirmation with a `task_id`, so callers parsing the old return text must be updated.

— names tool and exact old/new return shape v0.5.0

09 **CLI commands follow OPENVIKING_CLI_CONFIG_FILE** BREAKING 65

`ov status`, `ov config validate`, the config wizard, and `ov config switch` now follow the file selected by `OPENVIKING_CLI_CONFIG_FILE` instead of always targeting the default config file.

— names exact commands and env var v0.5.0

10 **show_recall field in context profiles** NEW 60

A new `show_recall` field on context profiles (off by default) makes the gateway's first reply each turn show a summary of what OpenViking injected; this summary is never sent to the model or stored in memory.

— names the field and behavior but no runnable example v0.5.0

11 **exclude_uris now excludes entire subtrees** BREAKING 60

`exclude_uris` now excludes an entire subtree when given a directory URI, instead of matching only the exact string; callers passing directory URIs that previously expected

no-op exact-match behavior will now see more results excluded.

— names parameter and behavior change, no endpoint given v0.5.0

THINNER COVERAGE BELOW

12 **--disable-tool flag for vikingbot chat** NEW 55

A repeatable `--disable-tool` flag on `vikingbot chat` lets users selectively disable tools at invocation time.

— names exact flag and command, thin on mechanism v0.5.0

13 **DSH session IDs normalized for Windows-illegal characters** 45

BREAKING

DSH session IDs containing Windows-illegal characters or trailing dots/spaces are now normalized with a hash suffix; affected sessions will map to new OpenViking sessions after upgrade.

— names affected integration but no exact mechanism detail v0.5.0

└──▷ BREAKING ON UPGRADE

! `memory_policy.working_memory.enabled` default changed from `true` to `false`; deployments that need Working Memory must re-enable it explicitly. Policies previously serialized with `working_memory` omitted (because it was `true`) will now be read as `false` after upgrade.

! Claude Code, Codex, DSH, OpenCode, and pi plugin commits now send `keep_recent_count: 0`; the `commitKeepRecentCount` setting and `OPENVIKING_COMMIT_KEEP_RECENT_COUNT` environment variable are no longer read.

! `/api/v1/search/search` in `mode: "context"` no longer returns system-preset directories (`viking://resources`, `viking://agent`, user root and its first-level directories); results may differ from v0.4.23.

! `exclude_uris` now excludes an entire subtree when given a directory URI instead of matching only the exact string; callers passing directory URIs that expected no-op exact-match behavior will see more results excluded.

! MCP `remember` no longer returns the fixed string `Stored N message(s) and committed for memory extraction.`; it now returns a submission confirmation with a `task_id`. Callers parsing the old return text must be updated.

! `ov status`, `ov config validate`, the config wizard, and `ov config switch` now follow the file selected by `OPENVIKING_CLI_CONFIG_FILE` instead of always targeting the default config file.

! DSH session IDs containing Windows-illegal characters or trailing dots/spaces are normalized with a hash suffix; affected sessions will map to new OpenViking sessions after upgrade.

! Claude Code, Codex, DSH, OpenCode, and pi (non-takeover mode) plugin integrations default to letting the host manage history and compaction rather than OpenViking; existing

explicit config and environment variables still take precedence.

WAS THIS USEFUL? 👍 👎

Docling

Feedback ⓘ v2.137.0 NOTES ↗ RANK

Docling parses PDFs, Office files, and HTML into structured documents with layout, tables, and reading order preserved for downstream RAG pipelines.

> Docling's biggest addition this window is post-VLM chart data extraction and picture description/classification enrichment, alongside new iWork, Word, and LaTeX backend options and improved heading hierarchy detection.

└──▷ **WHAT SHIPPED · 6 FEATURES** most completely described first ? what's the number?

01 Post-VLM chart extraction and picture enrichment NEW

95

Adds `--enrich-chart-extraction` CLI flag and VlmPipelineOptions(do_chart_extraction=True) Python option to extract chart data after VLM conversion, automatically enabling picture classification. The extraction model can be swapped via `--chart-extraction-preset granite_vision_v3_3` (ChartExtractionVlmEngineOptions.from_preset('granite_vision_v3_3')) for the older CSV-only `ibm-granite/granite-vision-3.3-2b-chart2csv-preview` model, or `--chart-extraction-preset granite_vision_v4_mlx` (from_preset('granite_vision_v4_mlx')) to force MLX execution on Apple Silicon with `ibm-granite/granite-vision-4.1-4b`. `--enrich-picture-description` and `--enrich-picture-classes` flags enable picture description and classification independently of chart extraction, made possible by new support for running enrichments after the VLM pipeline.

Extract structured chart data from an image after VLM conversion using the default Granite Vision 4.1 4B model.

```
$ docling --from image --to dclx IMAGE.png --vlm-model
mineru2_pro --enrich-chart-extraction
```

Extract chart data in Python, letting Docling automatically enable picture classification and skip fields already present in VLM output.

```python
from docling.pipeline.vlm_pipeline import VlmPipelineOptions

options = VlmPipelineOptions(do_chart_extraction=True)
# pass options to your DocumentConverter as pipeline_options
```

Force the older CSV-only chart model when you need legacy CSV output instead of the default Granite Vision 4.1 4B output.

```
$ docling --from image --to dclx IMAGE.png --vlm-model
mineru2_pro --enrich-chart-extraction --chart-extraction-preset
granite_vision_v3_3
```

— Runnable commands and named flags/models for every option. v2.137.0

02 **Markdown emphasis mapping from dots OCR output** NEW

70

Markdown bold/italic emphasis and inline HTML in native `dots.ocr` and `dots.mocr` JSON text are now mapped to styled text spans, preserving titles, headings, list items, formula blocks, inline math, and code; available via `pip install 'docling-slim[format-markdown]'`.

— Names install extra and preserved elements, no usage example. product docs

03 **iWork backend enhancements for Numbers and Keynote** NEW

65

The Numbers backend gains an optional `iwork_numbers` sheet-name filter list that restricts processing to sheets whose names appear in the list (case-sensitive, defaults to `None` for all sheets). Keynote presentations and Numbers spreadsheets also gain opt-in chart image rendering that rasterizes each chart via LibreOffice and attaches it to its `PictureItem`; it is off by default since it requires LibreOffice and increases output size.

— Named config field and behavior, lacks usage example. product docs

04 **Heading hierarchy inference improvements** IMPROVED

60

Heading hierarchy inference now recognizes Division/Subdivision heading markers, and the Word backend adds a `use_outline_level_for_headings` option that falls back to `w:outlineLvl` as a heading signal when no name-based style match exists, enabling correct detection for localized style names such as `Nadpis1`; set to `False` to disable.

— Named config option and behavior, no code example given. v2.136.0

THINNER COVERAGE BELOW

05 **LaTeX backend stream path resolution** NEW

45

The LaTeX backend adds a local path field so that when a document is read from a stream, relative `\input`, `\include`, and `\includegraphics` paths are resolved against the supplied directory.

— Brief mechanism, no example or config shown. product docs

06 **Repetition penalty options for MLX inference** NEW 25

Adds repetition penalty options for MLX model inference.

— Bare one-line addition, no detail on effect. v2.136.0

WAS THIS USEFUL? 👍 👎

EVALUATE

◆ AI Observability & Evals

UK AI Security Institute Inspect AI

Feedback ⓘ

0.3.278 CODE ↗

Inspect is the UK AI Security Institute's framework for large language model evaluations, covering datasets, solvers, scorers, and agentic eval tasks.

Adds sentinel monitors for tool calls, `EvalSpec.shards` field, task/sample `description` field, handoff() review, and per-model throughput rates in `inspect ctl`.

└──▷ GET THIS VERSION ▾

```
$ git clone --branch 0.3.278
https://github.com/UKGovernmentBEIS/inspect_ai.git
# already have the repo? check out this version:
$ git checkout 0.3.278
```

Adds experimental sentinel monitors and protocols for tool calls via the new `inspect_sentinel` integration (imports: `BeforeToolCall`, `AfterToolCall`, `Decision`, `Report`, `SentinelEvent`, `SentinelStatus`, `SentinelSuspicion`, `SentinelAction`).

Adds `EvalSpec.shards` field to merged shard logs, recording shard metadata in the eval log.

Lets `Task` and samples carry a `description` field, recorded in the eval log and surfaced in evals_df() and samples_df().

Extends tool review to also review handoff() calls, the same as approval already does.

Adds `inspect ctl model throughput` display of input and cache token rates per model.

NEW SUPPORTED PLATFORM Claude Haiku 5.5 support

└──▷ BREAKING ON UPGRADE

! Removes ModelAPI.apply_redacted_reasoning_tokens_to_input() and stops adding `redacted_reasoning_tokens` to assistant message metadata — code that calls this method or reads this field will break.

! Setting `text` on a `ChatMessage` with a list of content now replaces the existing text in place rather than appending after images; multimodal evals using

prompt_template(), chain_of_thought(), or multiple_choice() may see shifted baselines on rerun.

WAS THIS USEFUL? 👍 👎

Langfuse

Feedback ⓘ v4.56.0 NOTES ↗ RANK

Langfuse provides tracing, evaluation, and monitoring for LLM applications.

> Langfuse v4.56.0 adds spend alerts for ClickHouse-billed organizations, revamps the evals decision view, and improves scalability by routing several read types to a read replica.

└──▷ **WHAT SHIPPED · 5 FEATURES** most completely described first ? what's the number?

01 **Read replica routing for reads** IMPROVED 50

Score, trace, prompt, dataset, and session reads are now routed to the read replica, improving scalability for high-volume deployments.

– Names affected data types and stated scalability benefit v4.56.0

02 **Spend alerts for ClickHouse-billed orgs** NEW 25

Langfuse now sends spend alerts for organizations billed on ClickHouse usage.

– Named but no mechanism, threshold or config given v4.56.0

03 **Revamped decision results in evals view** IMPROVED 25

The evals view has revamped decision model results.

– No detail on what changed in the UI v4.56.0

04 **Adaptive map exploration in topics feature** IMPROVED 25

The topics feature gains adaptive map exploration and a compact workspace layout.

– Two additions named but no mechanism described v4.56.0

05 **Syntax highlighting in session tool previews** NEW 25

Session tool previews now have syntax highlighting.

– Bare name, no further detail v4.56.0

WAS THIS USEFUL? 👍 👎

Comet Opik

Feedback ⓘ 2.2.96 NOTES ↗ RANK

Opik traces, evaluates, and monitors LLM applications and agentic workflows, with datasets, an LLM-as-judge metric library, and production dashboards.

> Opik 2.2.96 ships a small UI improvement that keeps Logs page filters persistent across navigation.

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

01 **Persistent Logs filters across navigation** IMPROVED

33

The Logs page now retains active filter state when a user navigates away and returns, so filters no longer reset.

— Describes behaviour but no exact UI path or config detail 2.2.96

WAS THIS USEFUL? 👍 👎

Braintrust

Feedback ⓘ snapshot-20261010 NOTES ↗ RANK

Braintrust provides evaluation, tracing, and improvement workflows for AI applications.

> Braintrust shipped cooperative task cancellation hooks, span customizers for redacting exported trace data, and new custom-view versioning and blind-review controls in Human review, alongside several breaking changes to span metadata fields, a .NET tracing builder API, and zero-value timeout behavior.

└──▷ **WHAT SHIPPED · 6 FEATURES** most completely described first ? what's the number?

01 **Cooperative task cancellation via hooks** NEW 90

Task hooks now expose `hooks.cancel_event` and `hooks.signal` for cooperative cancellation inside evaluation tasks, with the `timeout` setting controlling both; a task can check hooks.cancel_event.is_set() to stop cleanly when the timeout fires.

Use cooperative cancellation in an evaluation task so it stops cleanly when a timeout fires.

python

```python
import braintrust

def my_task(input, hooks):
    for chunk in expensive_work():
        if hooks.cancel_event.is_set():
            return {'result': 'cancelled'}
        process(chunk)
    return {'result': 'done'}

braintrust.Eval(
    'my-project',
    data=cases,
    task=my_task,
    timeout=30,
)
```

— Runnable code example with named hook attributes product docs

02 **Custom view versioning and review controls** NEW 70

The Update saved view dialog in Human review now offers Save as new version and Save as new view options, letting teams version or branch a custom view without overwriting the original, plus a Revert changes option to discard unsaved edits to a custom view. Closing the Edit view editor with unsaved changes now prompts a Save to project confirmation to prevent accidental loss of edits. Custom views opened in

Human review also now enforce blind-review visibility rules, hiding peer answers and review spans while blind-review mode is active and keeping them hidden after scores are submitted.

Branch a shared custom view to experiment with layout changes without affecting the team's current saved view.

> 📍 In Human review, open the `Edit view` editor, make your changes, then select `Save changes` › `Save as new view` in the `Update saved view` dialog to create a separate view while leaving the original intact.

— Names each dialog option and mode but only a UI path to use it `product docs`

---

03 **Expanded span metadata for cache and agent tracing**

`UPDATED INTEGRATION` `IMPROVED` 63

Spans now record the `diagnostics` request parameter plus `cache_miss_reason` and `cache_missed_input_tokens` in metadata when Anthropic reports a prompt cache miss. Responses API spans now record `access_programs` in metadata, and Agents API sessions now trace computer use calls (with screenshots captured as attachments), browser authentication requests, and computer use approval items.

— Names several metadata fields but gives no steps to consume them `product docs`

---

04 **Breaking span metadata field changes** `UPDATED INTEGRATION` 63

`BREAKING`

On pi model call spans, `cost` and `metadata.provider_request` are replaced by top-level fields, and `skill` loads now use `tool_name: 'read'` and `tool_kind: 'skill'` instead of the earlier fields. Spans also no longer include `metadata.bt_daemon_version`.

— Names affected fields but no migration guidance beyond renames `product docs`

THINNER COVERAGE BELOW

---

05 **External ID support without run_id** `IMPROVED` 55

process_submission_result() / processSubmissionResult() no longer requires a `run_id` / `runId` when you pass an `external_id` / `externalId`.

— Names the functions and params but no usage example `product docs`

---

06 **Zero timeout now fails fast** `BREAKING` 40

A `timeout` value of `0` now times out immediately instead of having no effect, changing behavior for any evaluation configured with `timeout=0`.

— States the behavior change but no example of its impact `product docs`

NEW INTEGRATION pi 1.0 codemode tracing

NEW SUPPORTED PLATFORM Windows DPAPI credential storage for bt CLI

└──▷ ALSO FROM THESE RELEASES

Submit many evaluation cases at once using WorkflowBatchTask to reduce round-trips to the provider.

```python
from braintrust import Eval, WorkflowBatchTask, WorkflowBatchScorer

braintrust.Eval(
    'my-project',
    data=cases,
    task=WorkflowBatchTask(my_workflow),
    scores=[WorkflowBatchScorer(my_scorer)],
)
```

└──▷ BREAKING ON UPGRADE

! ChatClientBuilder.UseBraintrustTracing() has been removed; replace it with UseBraintrustLLMTracing() on `ChatClientBuilder`, and move UseBraintrustFunctionTracing() from `ChatClientBuilder` to `AIAgentBuilder`.

! On pi model call spans, `cost` and `metadata.provider_request` are replaced by top-level fields; `skill` loads now use `tool_name: 'read'` and `tool_kind: 'skill'` — update any queries that reference the earlier fields.

! Spans no longer include `metadata.bt_daemon_version`.

! A `timeout` value of `0` now times out immediately instead of having no effect.

snapshot-20261010 NOTES ↗ LATEST STABLE

Adds coding agent cost/usage dashboards, SQL charts, a new trace thread view, Loop on dashboards, MCP-based dataset building, and eval cancellation via `cancel_event`.

Adds Coding agent insights dashboard at Dashboards > Coding agent insights showing usage by user, API key, model, tool, and branch with cost breakdowns; requires data plane v2.13.0+.

Adds a daily Loop automation on Coding agent insights to find recurring problems and inefficiencies across team sessions.

Adds a new thread view that groups each trace turn into request, agent work, and final response, with tool calls nested under the LLM call that requested them and Debugger findings highlighted inline.

Adds SQL charts — build dashboard charts from a SQL query and switch between query text and a visual builder editing the same query; requires data plane v2.1.1+

and Pro/Enterprise plans.

Adds ability to chart query results in the SQL sandbox and save them directly to a dashboard.

+9 more

└─▷ **BREAKING ON UPGRADE**

! On pi model call spans (`bt` CLI v0.23.0), `reasoning_tokens` is renamed to `completion_reasoning_tokens` , `cost` is renamed to `estimated_cost` , `metadata.provider_request` is replaced by top-level fields such as `temperature` and `tools` , and skill loads now use `tool_name: 'read'` with `tool_kind: 'skill'` . Queries using the earlier field names must be updated.

! Spans no longer include `metadata.bt_daemon_version` (`bt` CLI v0.23.0). Queries or dashboards filtering on this field will return no results.

WAS THIS USEFUL? 👍 👎

GOVERN

◆ AI Security Posture & Discovery

Mend.io Mend AI

Feedback ⓘ changelog · Oct 10, 2026

CHANGELOG ↗

Mend AI is an AI security service that discovers AI components and helps teams govern their use in software applications.

UPDATED DETECTION Mend AI Runtime Protection | Guardrails multi-language support

WAS THIS USEFUL? 👍 👎

◆ AI/LLM Security

agent-safe-pipeline

Feedback ⓘ decionis-cli-v0.2.1

NOTES ↗ RANK

Agent Safe Pipeline provides a security-focused framework for building AI agent workflows with policy checks and safety controls.

> Agent Safe Pipeline's Decionis CLI 0.2.1 adds a self-contained Windows executable, removing the need for a separate Node.js or Bun install.

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

01 Standalone Windows executable for Decionis CLI NEW

82

Adds `decionis.exe`, a standalone Windows x64 executable built with Bun 1.4.2 that embeds the CLI, all dependencies, and the runtime, eliminating the need to install Node.js or Bun separately. It is packaged as a ZIP at `decionis/bin/decionis.exe` with Windows usage instructions, third-party notices, checksums, and WinGet manifests for PATH-based or package-manager installation.

After extracting the ZIP, add the binary to PATH so `decionis` is available system-wide on Windows without any runtime prerequisites.

```
$ Expand-Archive decionis-cli-v0.2.1.zip -DestinationPath
C:\tools
$env:PATH += ';C:\tools\decionis\bin'
```

— Names exact binary path, build tool, and install methods with a runnable command.

decionis-cli-v0.2.1

WAS THIS USEFUL? 👍 👎

NVIDIA OpenShell

Feedback ⓘ v0.1.3 NOTES ↗ RANK

OpenShell runs autonomous AI agents in policy-controlled sandboxes with isolated filesystem, network, and credential access.

> OpenShell v0.1.3 adds an OCI Generative AI provider profile, OCSF-formatted JSONL audit logging for the gateway, bearer token passthrough, and a cleaner sandbox detach workflow, alongside smaller additions to providers, Docker runtime, and observability.

└──▷ **WHAT SHIPPED · 11 FEATURES** most completely described first ? what's the number?

01 **Ctrl-D sandbox session detach** NEW 70

Pressing Ctrl-D inside a sandbox session now detaches cleanly rather than terminating it (`feat(cli): detach sandbox sessions with Ctrl-D`), letting a sandbox stay alive during CI or long-running agent workflows.

Detach from a running sandbox session without terminating it — useful in CI or long-running agent workflows where you want to leave the sandbox alive.

```
$ openshell sandbox connect my-sandbox
# Inside the session, press Ctrl-D to detach cleanly
```

— Includes a runnable command demonstrating the behavior v0.1.3

THINNER COVERAGE BELOW

02 **OCSF JSONL audit logging for gateway** NEW 58

The gateway now emits audit events in OCSF format to a JSONL file (`feat(server): write gateway OCSF events to JSONL`), and this output can be configured via new Helm values (`feat(helm): configure gateway OCSF JSONL output`).

— Names format and Helm surface, no exact key given v0.1.3

03 **Disable telemetry via environment variable** NEW 50

Anonymous telemetry collection on a self-hosted gateway can be disabled without recompiling by setting `OPENSHELL_TELEMETRY_ENABLED=false` before start-up.

Disable anonymous telemetry collection on a self-hosted gateway without recompiling — set this in your gateway environment before start-up.

```
$ OPENSHELL_TELEMETRY_ENABLED=false openshell sandbox create --
name demo
```

– Exact env var and command shown, but not in changelog bullets v0.1.3

04 **OCI Generative AI provider profile** NEW 47

A new Oracle Cloud Infrastructure Generative AI provider profile (`feat(providers): add Oracle Cloud Infrastructure Generative AI provider profile`) is available for credential-scoped inference routing.

– Names the provider but no config example v0.1.3

05 **Bearer authorization passthrough** NEW 45

The gateway can now forward bearer tokens directly to upstream endpoints without credential substitution (`feat(service): add bearer authorization passthrough`).

– Describes behavior but no command shown v0.1.3

06 **OTLP trace export from sandbox supervisors** NEW 45

Sandbox supervisors now emit OpenTelemetry traces over OTLP for observability pipelines (`feat(supervisor): export OTLP traces from sandbox supervisors`).

– Names OTLP export but no config/endpoint detail v0.1.3

07 **Sandbox main-process restart policy** NEW 40

Sandboxes now support a main-process restart policy to recover from agent crashes (`feat(sandbox): add main restart policy`).

– Names the policy but no config syntax given v0.1.3

08 **Docker corporate proxy CA bundle support** NEW 40

The Docker runtime now accepts corporate proxy CA bundles for TLS inspection environments (`feat(docker): support corporate proxy CA bundles`).

– Names the capability but no flag or config path v0.1.3

09 **MCP request inspection with Tower protocol profiles** NEW 30

MCP requests can now be inspected using Tower-selected protocol profiles (`feat(mcp): inspect requests with Tower-selected protocol profiles`).

– Thin description with no usage detail v0.1.3

10 **On-demand sandbox config file serving** IMPROVED 30

Providers can now serve sandbox configuration files on demand rather than bundling them at creation time (`feat(providers): serve sandbox config files on demand`).

— Describes change but no mechanism or syntax v0.1.3

11 **Multiple tokens per provider request** NEW 25

A single provider request can now carry multiple tokens (`feat(providers): add multiple tokens to one request`).

— Bare description with no example or field name v0.1.3

WAS THIS USEFUL? 👍 👎

AI MODELS

◆ Frontier Models

Anthropic

Feedback ⓘ snapshot-20261010 NOTES ↗ RANK

Anthropic provides Claude AI models and APIs for building applications that generate, analyze, and automate work.

> Anthropic shipped dynamic multi-agent workflow orchestration for Claude Managed Agents and added Word-file export for Claude Docs via the Compliance API.

└──▷ **WHAT SHIPPED · 2 FEATURES** most completely described first ? what's the number?

01 Dynamic workflow orchestration for Managed Agents NEW

95

Managed Agents gain a new `multiagent_20261001` agent type that enables dynamic workflows by default, letting an agent write and execute a workflow program that runs many agents in phases and combines their results as a background workflow run; this is accessed via the `managed-agents-2026-04-01` beta header. Which agents can be called as subagents or workflow participants is configured through `subagents.predefined_agents` and `workflows.predefined_agents` lists, an optional advisor model can run alongside the primary agent, and each run can be tracked via `workflow_run.*` events on the session's event stream. Sending `"multiagent": null` clears the entire multiagent config block and `"workflows": null` resets dynamic workflows to the default; otherwise config updates merge level-by-level, with omitted settings keeping their stored value while a sent `predefined_agents` list fully replaces the stored one. Dynamic workflows can be turned on explicitly with a `multiagent` field set to `{"type": "multiagent_20261001", "workflows": {"type": "enabled"}}`.

— Names exact config keys, field values, beta header and event stream snapshot-20261010

THINNER COVERAGE BELOW

02 Word export for Claude Docs via Compliance API NEW

45

The Compliance API adds beta support for downloading Claude Docs documents as Word files.

— Names API and format but no endpoint path or limits given snapshot-20261010

WAS THIS USEFUL? 👍 👎

Mistral AI Mistral Nemo

Feedback ⓘ

price increase · Oct 9, 2026 MODEL ↗

Mistral Nemo is a chat model from Mistral AI.

Price increase on OpenRouter (mistralai/mistral-nemo): $0.019 → $0.029 in, $0.03 → $0.03 out per million tokens (+53%), first seen 9 Oct 2026.

WAS THIS USEFUL? 👍 👎

◆ Open-Weight Models

DeepSeek V3

Feedback ⓘ price increase · Oct 9, 2026 MODEL ↗

DeepSeek V3 is a chat model from DeepSeek.

Price increase on OpenRouter (deepseek/deepseek-chat): $0.2574 → $0.32 in, $1.0287 → $0.89 out per million tokens (+24%), first seen 9 Oct 2026.

WAS THIS USEFUL? 👍 👎

DeepSeek V4 Pro 0813

Feedback ⓘ

price increase · Oct 9, 2026 MODEL ↗

DeepSeek V4 Pro 0813 is a chat model from DeepSeek.

Price increase on OpenRouter (deepseek/deepseek-v4-pro-0813): $0.3458 → $0.66 in, $1.0375 → $1.98 out per million tokens (+91%), first seen 9 Oct 2026.

WAS THIS USEFUL? 👍 👎

Z.ai GLM 5.3

Feedback ⓘ price cut · Oct 9, 2026 MODEL ↗

GLM 5.3 is a reasoning model from Z.ai.

Price cut on OpenRouter (z-ai/glm-5.3): $0.07 → $0.039 in, $7.00 → $4.80 out per million tokens (-44%), first seen 9 Oct 2026.

WAS THIS USEFUL? 👍 👎

Moonshot AI Kimi K2.6

Feedback ⓘ price cut · Oct 9, 2026 MODEL ↗

Kimi K2.6 is a multimodal model from Moonshot AI.

Price cut on OpenRouter (moonshotai/kimi-k2.6): $0.95 → $0.465 in, $4.00 → $2.45 out per million tokens (-51%), first seen 9 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3 235B A22B Thinking 2507

Feedback

ⓘ context change · Oct 9, 2026 MODEL ↗ STABLE

Qwen3 235B A22B Thinking 2507 is a reasoning model from Alibaba.

Context window shrank from 131K to 128K tokens on OpenRouter (qwen/qwen3-235b-a22b-thinking-2507), first seen 9 Oct 2026.

▾ 1 more release in this issue · 2026-10-09

> price increase · Oct 9, 2026 MODEL ↗ STABLE
>
> Price increase on OpenRouter (qwen/qwen3-235b-a22b-thinking-2507): $0.23 → $0.45 in, $2.30 → $3.50 out per million tokens (+96%), first seen 9 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3 30B A3B Feedback ⓘ

context change · Oct 9, 2026 MODEL ↗

Qwen3 30B A3B is a chat model from Alibaba.

Context window shrank from 131K to 41K tokens on OpenRouter (qwen/qwen3-30b-a3b), first seen 9 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3 VL 30B A3B Thinking Feedback ⓘ

price cut · Oct 9, 2026 MODEL ↗

Qwen3 VL 30B A3B Thinking is a reasoning model from Alibaba.

Price cut on OpenRouter (qwen/qwen3-vl-30b-a3b-thinking): $0.20 → $0.29 in, $2.40 → $1.00 out per million tokens (-58%), first seen 9 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3 VL 8B Instruct Feedback ⓘ

price increase · Oct 9, 2026 MODEL ↗

Qwen3 VL 8B Instruct is a multimodal model from Alibaba.

Price increase on OpenRouter (qwen/qwen3-vl-8b-instruct): $0.117 → $0.25 in, $0.455 → $0.75 out per million tokens (+114%), first seen 9 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.5 397B A17B

Feedback ⓘ

price increase · Oct 9, 2026 MODEL ↗

Qwen3.5 397B A17B is a multimodal model from Alibaba.

Price increase on OpenRouter (qwen/qwen3.5-397b-a17b): $0.45 → $0.55 in, $3.00 → $3.50 out per million tokens (+22%), first seen 9 Oct 2026.

WAS THIS USEFUL? 👍 👎

Alibaba Qwen3.5-35B-A3B

Feedback ⓘ

price cut · Oct 9, 2026 MODEL ↗

Qwen3.5-35B-A3B is a multimodal model from Alibaba.

Price cut on OpenRouter (qwen/qwen3.5-35b-a3b): $0.15 → $0.08 in, $1.00 → $0.75 out per million tokens (-47%), first seen 9 Oct 2026.

WAS THIS USEFUL? 👍 👎

// END OF TRANSMISSION

The AI Toolchain · curated daily · October 10, 2026
Releases & features from the open-source and commercial AI tools that practitioners actually run.

▷ Subscribe Feedback on this issue