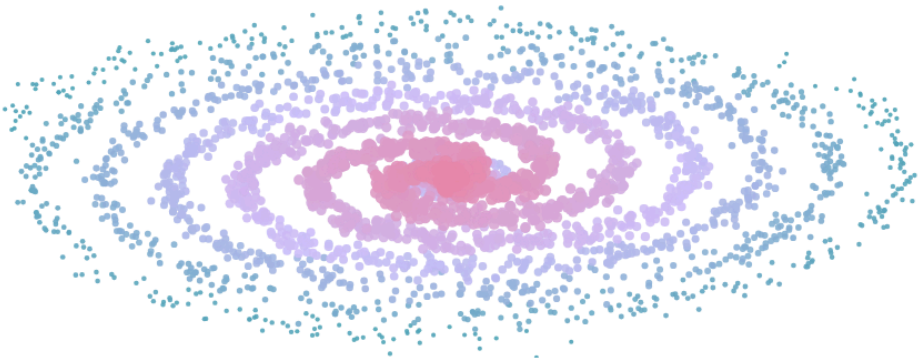
Bringing paper to life: A modern template for scientific writing



Publish-ready workflow that lets you focus on ideas, not infrastructure

AUTHOR

Thibaud Frere

AFFILIATION

Hugging Face

PUBLISHED

Sep. 01, 2025

DOI

10.1234/abcd.efgh

Table of Contents

Welcome to this open source research article template. It helps you publish clear, modern, and interactive technical writing with minimal setup.

Grounded in up to date good practices in web dev, it favors interactive explanations, clear notation, and inspectable examples over static snapshots.

Available on GitHub deployable on Hugging Face Spaces.

FEATURES

Markdown-based | KaTeX math | Syntax highlighting | Academic citations | Footnotes | Table of contents | Mermaid diagrams | Plotly ready | D3.js ready | HTML embeds | Gradio app embeds | Dataviz color palettes | Optimized images | Lightweight bundle | SEO friendly | Automatic build | Automatic PDF export | Dark theme | Mobile friendly | Latex import | Template update system

Introduction

The web offers what static PDFs can't: interactive diagrams, progressive notation, and exploratory views that show how ideas behave. This template treats interactive artifacts—figures, math, code, and inspectable experiments—as first-class alongside prose, helping readers build intuition instead of skimming results.

Who is this for

Ideal for anyone creating web-native and interactive content with minimal setup:

- For scientists writing modern web-native papers
- For educators building explorable lessons.

No web knowledge required—just write in Markdown.

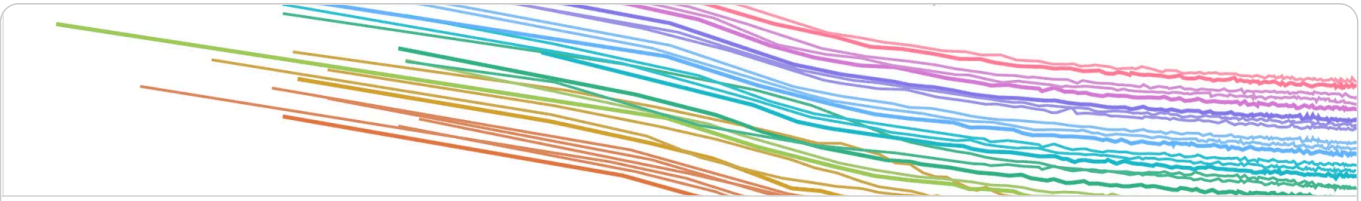
This is not a CMS or a multi-page blog—it's a focused, single-page, MDX-first workflow.

Inspired by Distill

This project stands in the direct continuity of Distill (2016–2021). Our goal is to carry that spirit forward and push it even further: accessible scientific writing, high-quality interactive explanations, and reproducible, production-ready demos.

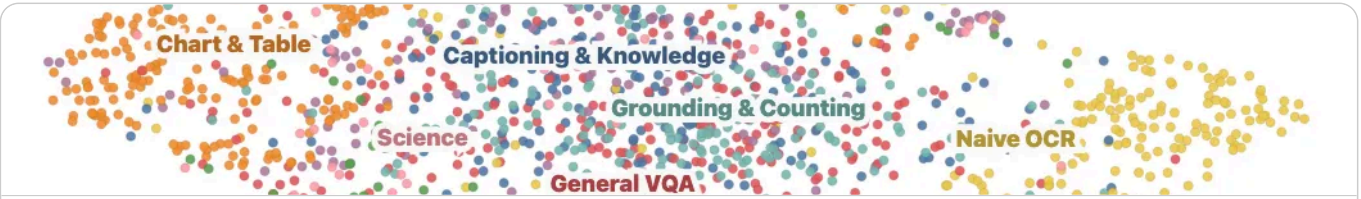
Built with this template

You can see how the template is used in the following examples. Discover more in the Research Article Gallery.



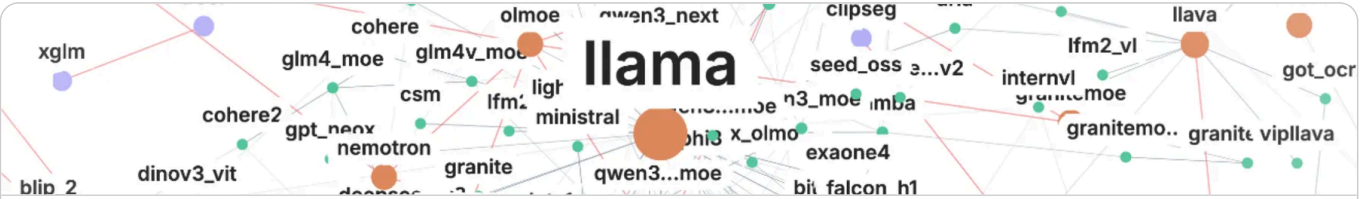
The Smol Training Playbook: The Secrets to Building World-Class LLMs

A practical journey through the challenges, decisions, and messy reality behind training state-of-the-art language models



FineVision: Open Data Is All You Need

A new open dataset for data-centric training of Vision Language Models



Maintain the unmaintainable: 1M python loc, 400+ models

A peek into software engineering for the transformers library

Your article?

Build your research paper with this template

Getting Started

Installation

The fastest way to get started is with npx:

```
npx create-research-article my-paper
```

You can also pick a template directly:

```
npx create-research-article my-paper --template=article
npx create-research-article my-paper --template=paper
```

The CLI guides you through the setup: project name, metadata (title, authors, affiliations), and template choice. It also offers to create and deploy a Hugging Face Space automatically if you have `huggingface-cli` installed.

Once done:

```
cd my-paper/app
npm run dev
```

You're ready to go! 🎉

> Requires Node.js 20 or newer. To manage versions, consider nvm (macOS/Linux) or nvm-windows.

▶ Alternative: duplicate the Space on Hugging Face

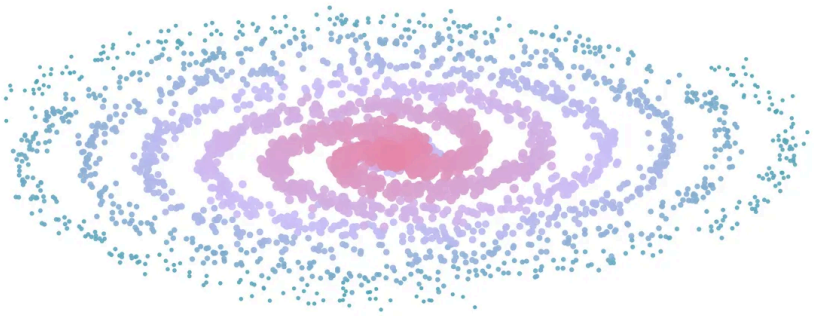
Template variants

You can choose the layout during `npx create-research-article`, or change it later in `article.mdx` frontmatter:

```
template: "article"  # full: banner, sidebar TOC, figure numbering, citation block, PDF export
template: "paper"    # lighter: centered single column, no figure numbering
```

See them live: article template | paper template

Exploring Self-Supervised Learning for Robotic Manipulation



A survey of recent advances in self-supervised learning applied to robotic manipulation tasks

AUTHORS
Thibaud Frere, Alice Martin

AFFILIATION
Hugging Face

PUBLISHED
Apr. 04, 2026

PDF
Download PDF

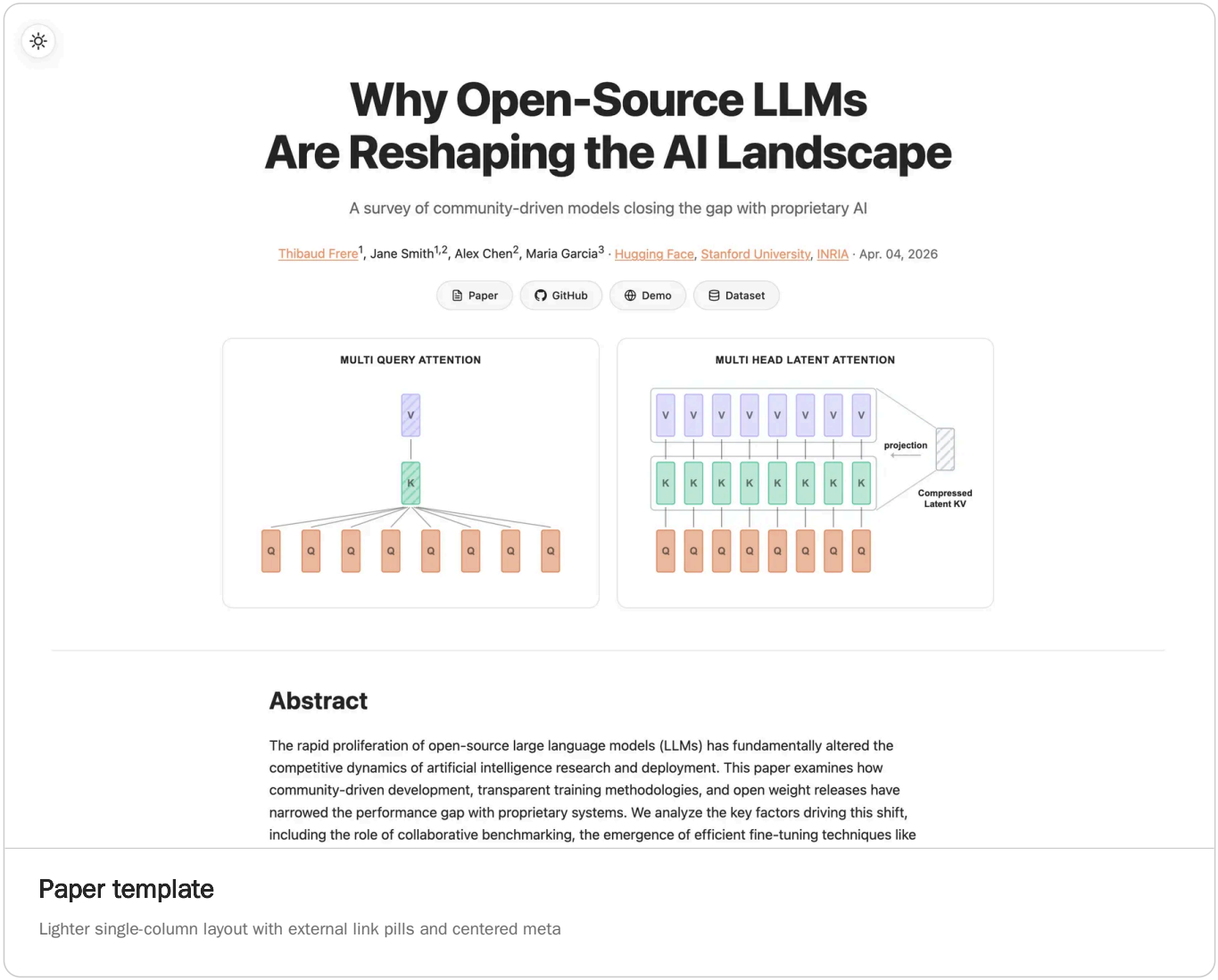
Table of Contents

Introduction

Recent years have witnessed a paradigm shift in how robots learn to interact with the physical world. Traditional approaches to robotic manipulation relied heavily on hand-engineered features, carefully calibrated controllers, and extensive domain expertise. While effective in structured environments, these methods struggle to generalize across the wide variety of objects, surfaces, and tasks that robots encounter in real-world settings.

Article template

Full layout with banner, sidebar TOC, figure numbering, citation block, and PDF export



Paper template

Lighter single-column layout with external link pills and centered meta

The `paper` template also supports external links (Paper, Code, Demo, Data…) displayed as pill buttons below the authors:

```yaml
links:
  - label: "Paper"
    url: "https://arxiv.org/abs/..."
  - label: "Code"
    url: "https://github.com/..."
```

Long titles are automatically balanced and downsized. You can force a line break with `\n` in the frontmatter:

```yaml
title: "Why Open-Source LLMs\nAre Reshaping the AI Landscape"
```

Development

```bash
npm run dev
```

Once started, the dev server is available at `http://localhost:4321`.

Build

```
npm run build
```

Deploy

Push to your Hugging Face Space to trigger an automatic build and deploy:

```
git add .
git commit -m "Update content"
git push space main
```

If you haven't set up a Space yet:

```
# create a Docker Space at huggingface.co/new-space, then:
git remote add space git@hf.co:spaces/<your-username>/<your-space>
git push space main
```

> Serving the `dist/` directory on any static host also works.

> A [slugified-title].pdf and thumb.jpg are generated at build time in the public folder.

Template Synchronization

Keep your project up-to-date with the latest template improvements. The sync system fetches the most recent changes from the official template repository at `https://huggingface.co/spaces/tfrere/research-article-template` and copies them to your project.

```
# Preview what would be updated
npm run sync:template -- --dry-run

# Update template files (preserves your content)
npm run sync:template
```

What gets preserved:

- Your content in `/src/content/`

What gets updated:

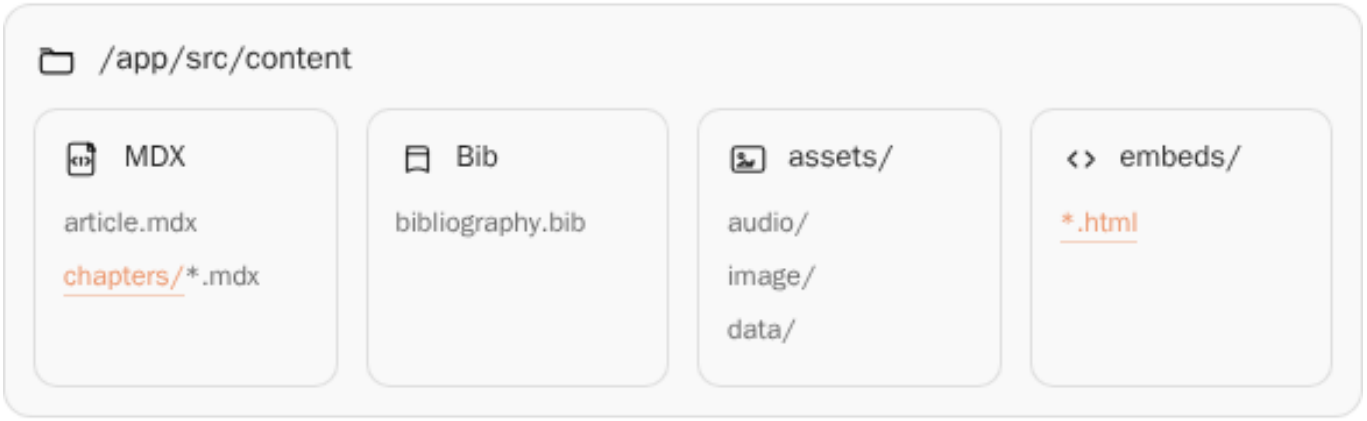
- All template files (components, styles, configuration)
- Dockerfile and deployment configuration
- Dependencies and build system

Writing your content

Once you have set up your project and started the development server, you can start writing your article.

Content structure

Your article lives in one and unique place. The `content` folder.



Article.mdx

The `article.mdx` file is the main entry point of your article which contains 2 main parts.

> MDX is a mix of Markdown and HTML/JSX: write regular Markdown and Components when needed. We'll describe the available options you can use later in this guide.

FRONTMATTER

Metadata and options for the article. Each of them is described in the table below.

Frontmatter in app/src/content/article.mdx

```yaml
---
title: "This is the main title"
subtitle: "This will be displayed just below the banner"
description: "A modern, MDX-first research article template with math, citations, and interactive figures."
published: "Feb 19, 2025"
tags:
  - research
  - template
authors:
  - name: "Thibaud Frere"
    url: "https://huggingface.co/tfrere"
    affiliations: [1]
  - name: "Alice Martin"
    url: "https://example.com/~alice"
    affiliations: [1, 2]
  - name: "Robert Brown"
    url: "https://example.com/~bob"
    affiliations: [2]
affiliations:
  - name: "Hugging Face"
    url: "https://huggingface.co"
  - name: "Example University"
    url: "https://example.edu"
doi: 10.1234/abcd.efgh
licence: Diagrams and text are licensed under <a href="https://creativecommons.org/licenses/by/4.0/" target="_blank" rel="noopener noreferrer">CC-BY 4.0</a> with the source available on <a href="https://huggingface.co/spaces/stfrere/research-article-template">Hugging Face</a>, unless noted otherwise. Figures reused from other sources are excluded and marked in their captions ("Figure from …").
seoThumbImage: "https://example.com/thumb.png"
tableOfContentsAutoCollapse: true
pdfProOnly: false
showPdf: true
---
```

Frontmatter fields

CONTENT

Your story. Write your content here.

Content in app/src/content/article.mdx

```mdx
import placeholder from '../../assets/image/placeholder.png'
import Image from '../../../components/Image.astro'
import Sidenote from '../../../components/Sidenote.astro'

This paragraph is written in Markdown.

<Sidenote>
  A short callout inserted via a component.
</Sidenote>
<Image src={placeholder} alt="Sample image with optimization" />

This paragraph is also written in Markdown.
```

CHAPTERS

If your article becomes too long for one file, you can organize it into separate chapters.

Simply create a new file in the `app/src/content/chapters` directory.
Then, include your new chapter in the main `article.mdx` like below.

Example

```
import MyChapter from './chapters/my-chapter.mdx';

<MyChapter />
```

You can see a living example here app/src/content/chapters/best-practices.mdx.

Table of contents

The Table of contents is generated automatically from your H2–H4 headings. Keep headings short and descriptive; links work on desktop and mobile.

> 💡 You can make the table of contents collapse by changing the `tableOfContentsAutoCollapse` parameter in the frontmatter. Which is `true` by default.

Theme

All interactive elements (buttons, inputs, cards, etc.) are themed with the primary color you choose.

You can update this main color to match your brand by changing the `--primary-color` variable in the `app/src/styles/_variables.css` file.

Use the color picker below to see how the primary color affects the theme.

BRAND COLOR

Aerospace Orange
OKLCH 75.0%, 12.0%, 47°
RGB 236, 149, 106
#EC956A

HUE

47°

COLOR PALETTES

Here is a suggestion of color palettes for your data visualizations that align with your brand identity. These palettes are generated from your `--primary-color`.

Use color with care. Color should rarely be the only channel of meaning. Always pair it with text, icons, shape or position. The simulation helps you spot palettes and states that become indistinguishable for people with color-vision deficiencies.

USING THE PALETTES

You can copy them manually from the palette viewer just above, or fetch colors via `window.ColorPalettes.getColors(key, count)` where `key` is one of `'categorical'`, `'sequential'`, `'diverging'`, and `count` is the desired number of colors (defaults to 6).

Code example

Markdown

All the following markdown features are available natively in the `article.mdx` file. No imports needed, just write markdown directly:

Text formatting — `**Bold**` → Bold, `*italic*` → *italic*, `~~strikethrough~~` → ~~strikethrough~~

Code — `` `inline code` `` → `inline code`, triple backticks for code blocks

Lists — `- Item` for bullets, `1. Item` for numbered lists with nesting support

Links — `[text](url)` → External links and internal navigation

Highlight — `<mark>text</mark>` → Highlighted text for emphasis

See also the complete Markdown documentation.

Advanced features — Explore specialized content types:

Math LaTeX equations	Code Syntax highlighting	Citation Academic references	Footnote Additional context
Mermaid Diagrams & flowcharts	Separator Visual breaks	Table Data presentation	Audio Sound embeds

Math

KaTeX provides full LaTeX math support with two simple syntaxes:

Inline math — Use `$...$` for equations within text: $x^2 + y^2 = z^2$

Block math — Use `$$...$$` for centered equations:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Advanced features — Aligned equations with IDs for cross-referencing:

$$\begin{aligned}
\log p_\theta(\mathcal{D}) &= \log \sum_{i=0}^{N} p_\theta((o,a)_i) && (1)\\
&= \log \sum_{i=0}^{N} \int_{\text{supp}(Z)} p_\theta((o,a)_i|z)p(z) && (2)\\
&= \log \sum_{i=0}^{N} \int_{\text{supp}(Z)} \frac{q_\theta(z|(o,a)_i)}{q_\theta(z|(o,a)_i)} \cdot p_\theta((o,a)_i|z)p(z) && (3)\\
&= \log \sum_{i=0}^{N} \mathbb{E}_{z\sim p_\theta(\bullet|(o,a)_i)}\left[\frac{p(z)}{q_\theta(z|(o,a)_i)} \cdot p_\theta((o,a)_i|z)\right], && (4)
\end{aligned}$$

You can reference equations with links like this equation.

Code example

Code

Use inline code with backticks `` `…` `` or ```` ``` ```` fenced code blocks ```` ``` ```` with a language for syntax highlighting (e.g., `` `python` ``).

As an example, here is inline code: `greet("Astro")` and below is a block.

```python
def greet(name: str) -> None:
    print(f"Hello, {name}!")
```

Code example

Code output

If you want to display the output of a code block, you can use the `:::output` directive. If it's directly below the code block, it will adapt to the code block's styling.

```python
def greet(name: str) -> None:
    print(f"Hello, {name}!")

greet("Astro")
```

OUTPUT

```
Hello, Astro!
```

Or it can also be used at a standalone block.

OUTPUT

```
Hello i'm a standalone output block.
```

It also works in an accordion

Code example

Citation

The citation keys come from `app/src/content/bibliography.bib`.

Citation use the `@` syntax (e.g., `[@vaswani2017attention]` or `@vaswani2017attention` in narrative form) and are automatically collected to render the bibliography at the end of the article.

1. In-text citation with brackets: (Vaswani et al., 2017).
2. Narrative citation: As shown by Kingma & Ba (2015), stochastic optimization is widely used.
3. Multiple citations and a footnote together: see (He et al., 2016; McKinney, 2017) for related work.
4. All citations in one group: (Cover & Thomas, 2006; Doe, 2020; He et al., 2016; Kingma & Ba, 2015; McKinney, 2017; OpenAI, 2023; Pedregosa et al., 2024; Raffel et al., 2020; Silver et al., 2017; E. Smith et al., 2024; J. Smith et al., 2021; Vaswani et al., 2017).

Code example

You can change the citation style in the `astro.config.mjs` file. There are several styles available: `apa`, `vancouver`, `harvard1`, `chicago`, `mla`. Default is `apa`.

Footnote

Footnote use an identifier like `[^f1]` and a definition anywhere in the document, e.g., `[^f1]: Your explanation`. They are numbered and listed automatically at the end of the article.

1. Footnote attached to the sentence above [1].

2. Multi-paragraph footnote example [2].

2. Footnote containing a list [3].

3. Footnote with an inline code and an indented code block [4].

4. Footnote that includes citation inside [5] and another footnote [1].

5. Footnote with mathematical expressions [6].

Code example

Referencing

In research articles, you may have to make references to anything. They are basically html anchors. They can be used internally in the article or externally in other articles.

1. Title
   Each title is automatically generated with a slugged version from the citation key. (slugged title from the citation key) like for example, the id `#mermaid-diagrams` is generated from the `Mermaid diagrams` title.

   Example Mermaid diagrams

2. Image and chart
   You can make a link to an image or a chart by adding an ID on it.
   `<HtmlEmbed id="neural-network-mnist-like"/>` then you can link to it with a link like
   `<a href="#neural-network-mnist-like">Fig 1</a>`.

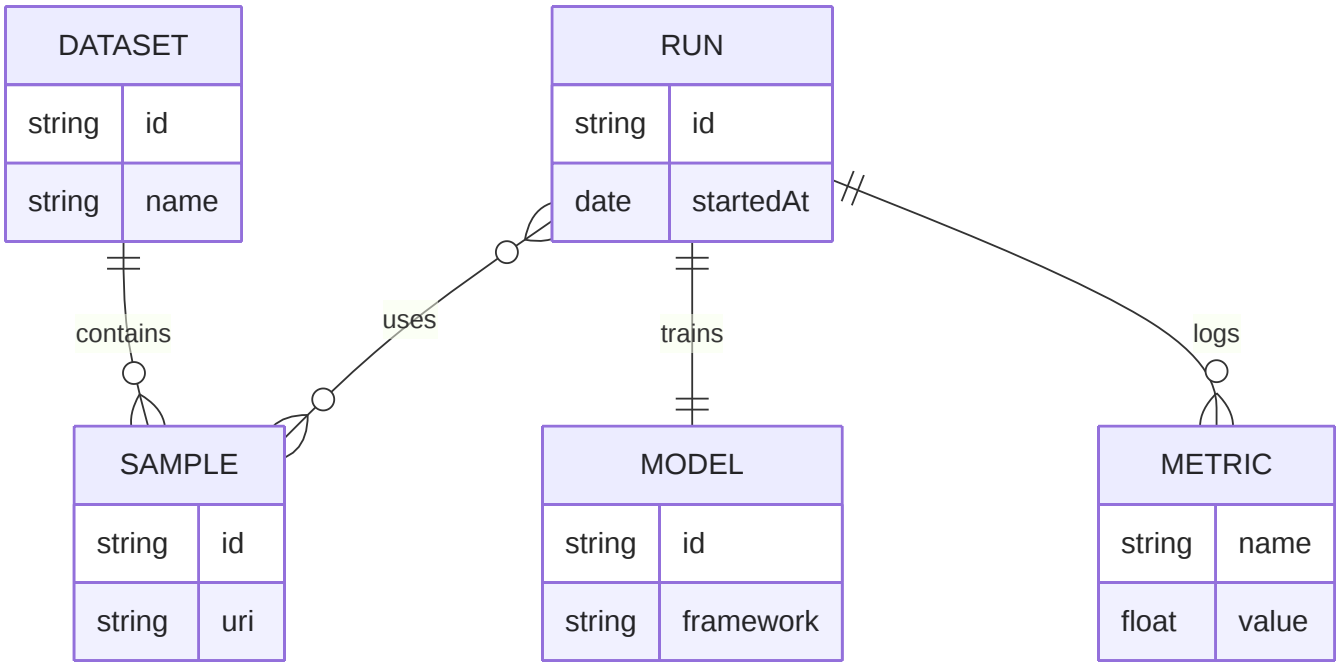
   Example Chart 1 or Fig 1

Available with: `Reference`, `Image`, and `HtmlEmbed` components all support the `id` prop for creating referenceable anchors.

Code example

Mermaid diagram

Native mermaid diagrams are supported (use a ```mermaid``` code fence). You can use the live editor to create your diagram and copy the code to your article.



Code example

Separator

Use `---` on its own line to insert a horizontal separator between sections. This is a standard Markdown “thematic break”. Don’t confuse it with the `---` used at the very top of the file to delimit the frontmatter.

Code example

Table

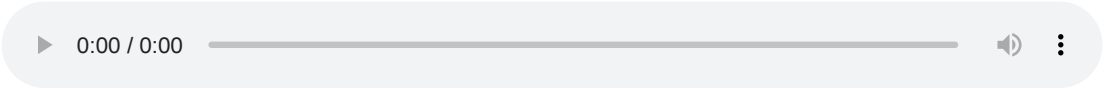
Use pipe tables like `| Column |` with header separator `| --- |`. You can control alignment with `:---` (left), `:---:` (center), and `---:` (right).

Model	Accuracy	F1-Score	Training Time	Status
BERT-base	0.89	0.89	2.5h	✅
RoBERTa-large	0.92	0.92	4.2h	✅
DeBERTa-v3	0.94	0.94	5.8h	✅
GPT-3.5-turbo	0.91	0.91	0.1h	✅

Code example

Audio

Embed audio using `<audio controls src={...} />`.



Code example

Components

All the following components are available in the article.mdx file. You can also create your own components by creating a new file in the `/components` folder.

How to import components

To use any component in your MDX file, add the import statement at the top:

```
import Image from '../components/Image.astro';
import Note from '../components/Note.astro';

# Your content

<Image src={myImage} alt="Description" />
<Note>This is a note</Note>
```

Here are the components that are available:

Image Optimized images	Placement Layout helpers	Accordion Collapsible content	Note Contextual callouts
Quote Elegant citations	HtmlEmbed External content	Reference Flexible content wrapper	Iframe Web embeds
Glossary Interactive term definitions	Stack Flexible grid layouts		

Image

Images automatically generate optimized `srcset` and `sizes` so the browser downloads the most appropriate file for the current viewport and DPR. You can also request multiple output formats (e.g., AVIF, WebP, fallback PNG/JPEG) and control lazy loading/decoding for better performance.

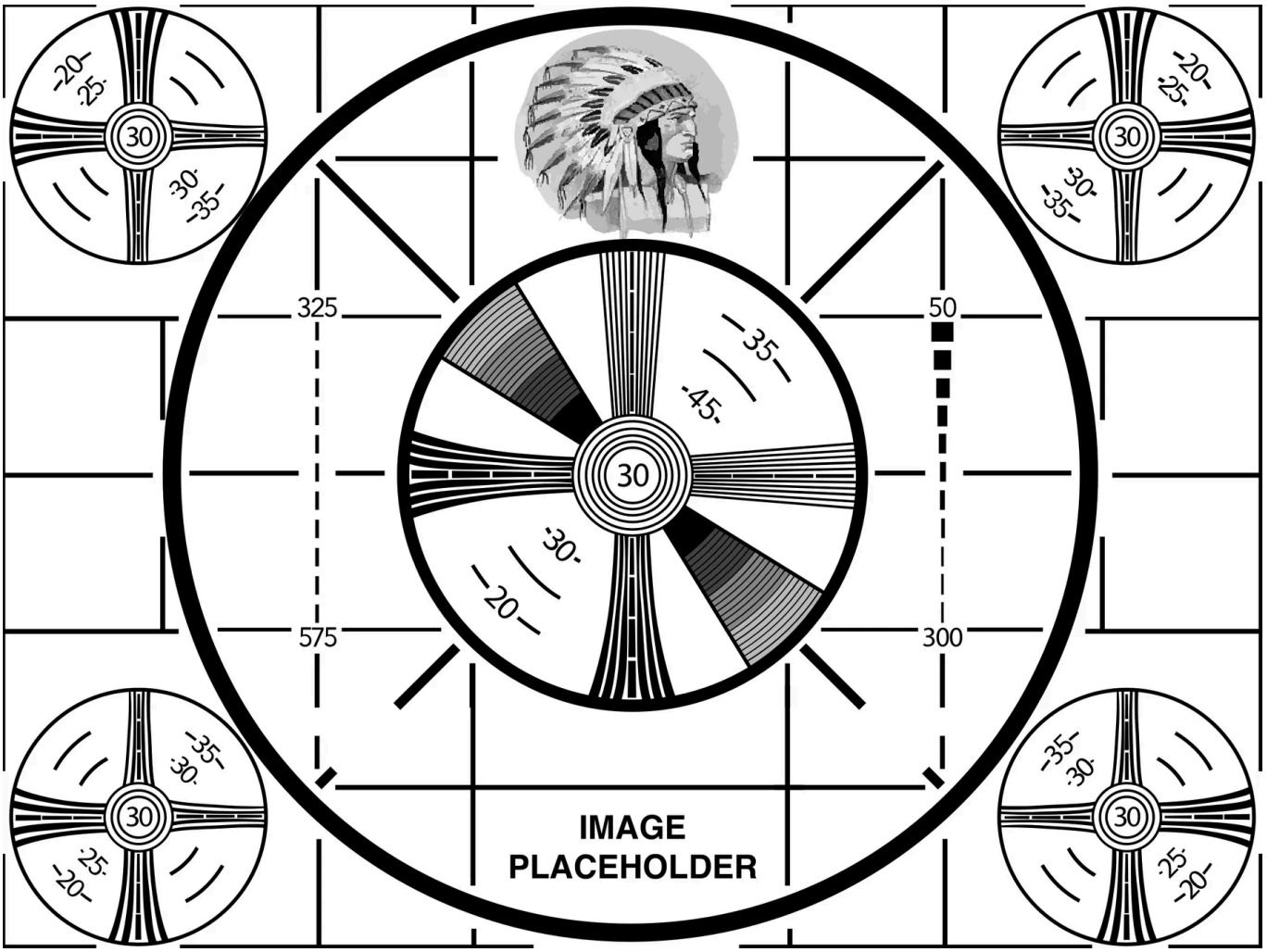


Figure 2 · A placeholder image description
Credit: RCA Indian Head Test Pattern

Prop	Required	Description
`zoomable`	No	Adds a zoomable lightbox (Medium-like).
`downloadable`	No	Adds a download button to fetch the image file.
`loading="lazy"`	No	Lazy loads the image.
`caption`	No	Adds a caption and credit.
`id`	No	Adds an `id` to the outer figure for deep-linking and cross-references.

Code example

Placement

Use these helpers when you need to step outside the main content flow: Sidenotes for contextual side notes, Wide to extend beyond the main column, and Full-width for full-width, immersive sections.

SIDENOTES

This paragraph presents a key idea concisely.

This paragraph discusses the relationship between variables in a mathematical context, where we consider the function $f(x) = x^2 + 2x + 1$.

Code example

WIDE EXAMPLE

demo wide

Code example

FULL-WIDTH EXAMPLE

demo full-width

Code example

Reference

The Reference component provides a flexible wrapper for any content with an ID and HTML caption. It's perfect for creating numbered references, cross-references, or any content that needs to be referenced throughout your article.

Note: The `Image` and `HtmlEmbed` components also support the same `id` and `caption` props for consistent referencing across your article.

you content here…

Figure 3 · Example reference with HTML caption and *styling*

Prop	Required	Type	Description
`id`	Yes	string	Unique identifier for the reference (used for deep-linking)
`caption`	Yes	string	HTML caption displayed below the content (supports HTML tags)

Code example

Use cases:

- Numbered figures with custom content
- Cross-references that can be linked to from anywhere in the article
- Flexible content blocks that need consistent styling and captioning
- Tables, charts, or any content that requires an ID and description

Accordion

Can be used like this `<Accordion>some content</Accordion>`. You can pass any children content.

What is an accordion?

The accordion component provides a collapsible content area that helps organize information efficiently. It's perfect for creating expandable sections, FAQ entries, or any content that benefits from progressive disclosure. Users can click the title to toggle the visibility of the content inside.

A table inside an accordion

Code example

Note

Small contextual callout for tips, caveats, or emphasis.

> 💡 Heads-up
>
> Use notes to surface context without breaking reading flow.

> Operation completed successfully.

> Be careful: this action cannot be undone.

> Plain note without header. Useful for short clarifications.

Prop	Required	Type	Description
`title`	No	string	Short title displayed in header
`emoji`	No	string	Emoji displayed before the title
`class`	No	string	Extra classes for custom styling
`variant`	No	'neutral'	'info'

Code example

Quote

Elegant quotes with optional source attribution.

> Backpropagation allows neural networks to discover their own internal representations of data.
>
> *Geoffrey Hinton, Learning representations by back-propagating errors*

Prop	Required	Type	Description
`source`	No	HTML	Quote source/author (supports HTML links)

Code example

Glossary

The Glossary component creates interactive term definitions with hover tooltips. Perfect for technical terms, acronyms, and concepts that need explanation without breaking the reading flow.

Machine Learning Neural Network API

Prop	Required	Type	Description
`term`	Yes	`string`	The word or term to define
`definition`	Yes	`string`	The definition of the term
`class`	No	`string`	Optional CSS class to apply to the term
`style`	No	`string`	Optional inline style to apply to the term
`position`	No	`'top' | 'bottom' | 'left' | 'right'`	Tooltip position (default: 'top')
`delay`	No	`number`	Delay before showing tooltip in ms (default: 30
`disableOnMobile`	No	`boolean`	Disable tooltip on mobile devices (default: fals

Code example

Use cases:

- Technical terms that need explanation
- Acronyms and abbreviations
- Domain-specific concepts for broader audiences
- Interactive glossaries for educational content

Stack

The Stack component provides flexible grid layouts for organizing content. Perfect for comparisons, side-by-side examples, multi-column layouts, and responsive content organization.

AUTO LAYOUT

This layout automatically adjusts based on content and available space.

RESPONSIVE

Items will wrap to new lines as needed, with a minimum width of 300px.

FLEXIBLE

Perfect for varying content lengths and dynamic layouts.

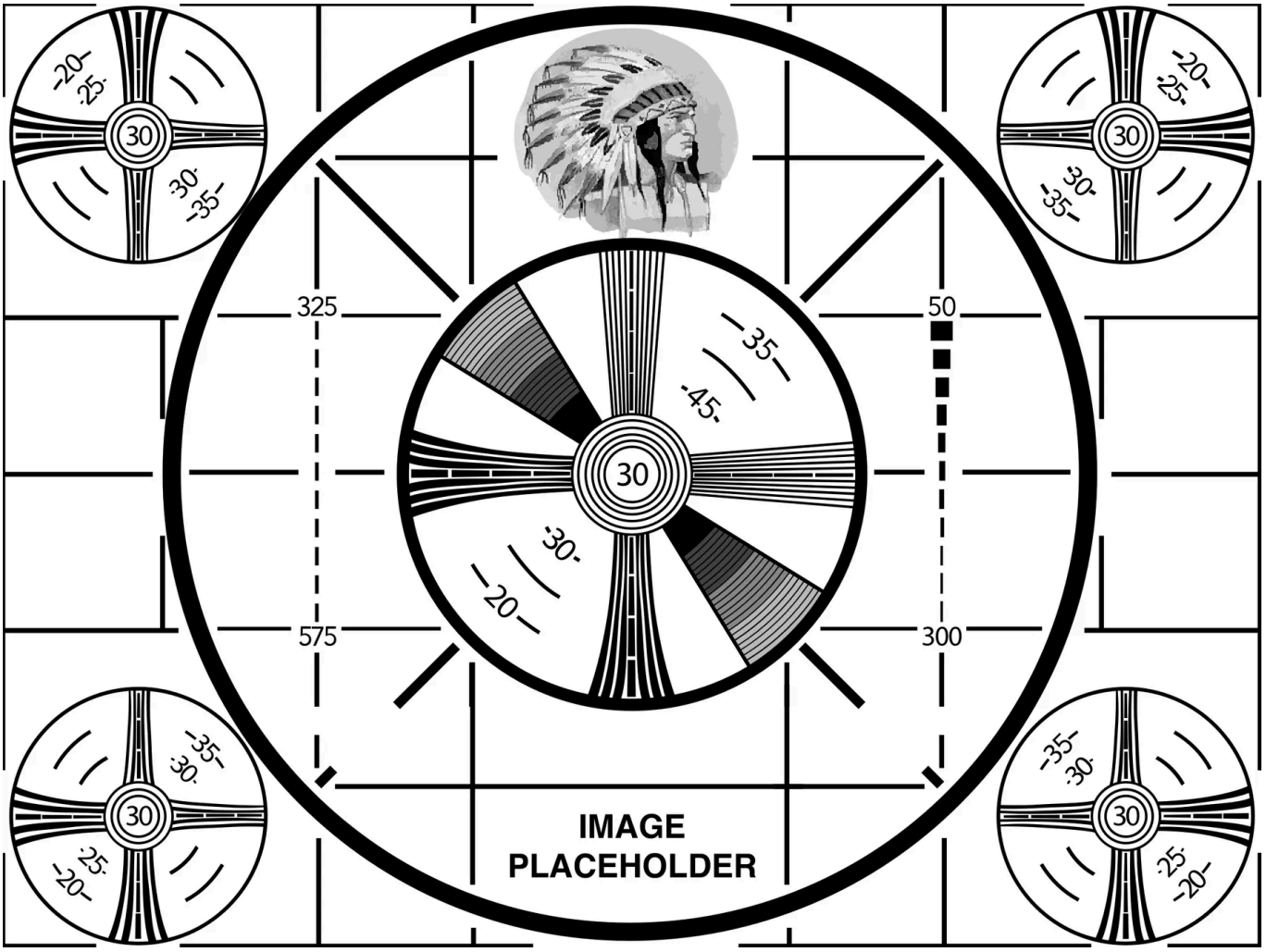


Figure 4 · Image 1: Example of using Image component within Stack

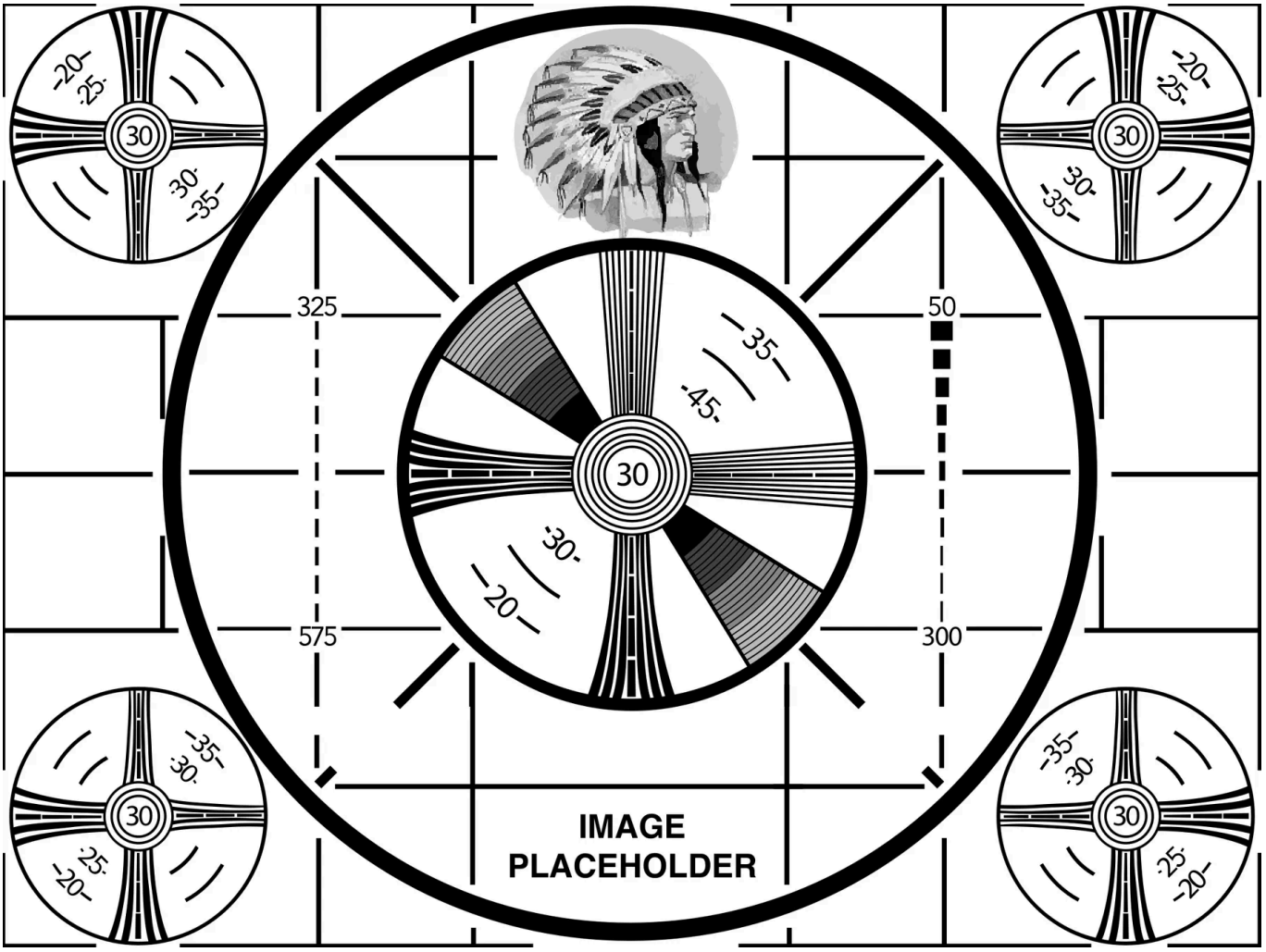


Figure 5 · Image 2: Side-by-side comparison layout

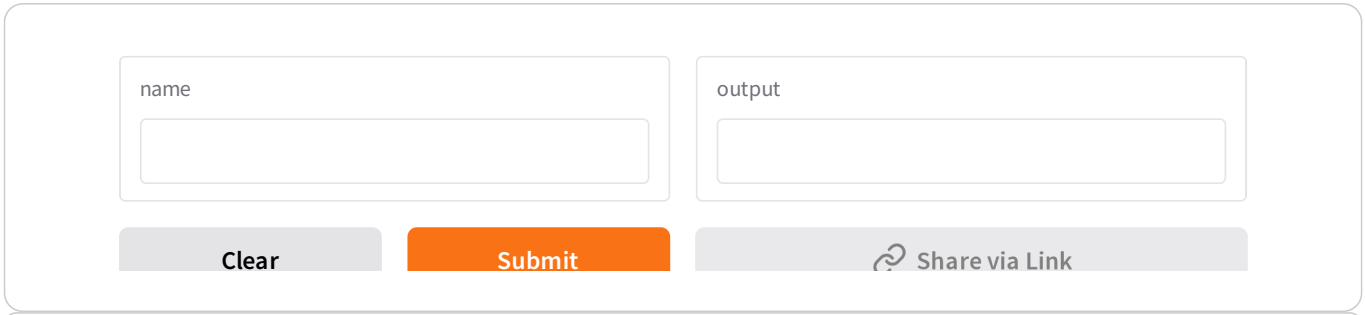
Prop	Required	Type	Description
`layout`	No	`"2-column" | "3-column" | "4-column" | "auto"`	Grid layout for the content (default: “2-columr
`gap`	No	`"small" | "medium" | "large" | string`	Gap between items - predefined size or custo
`class`	No	`string`	Optional CSS class to apply to the wrapper
`id`	No	`string`	Optional ID for the stack

Code example

Iframes

You can embed external content in your article using iframes. For example, Gradio or even Github code embeds can be used this way.

Gradio embed example



Code example

HtmlEmbed

The main purpose of the `HtmlEmbed` component is to embed a Plotly or D3.js chart in your article. Libraries are already imported in the template.

They exist in the `app/src/content/embeds` folder.

For researchers who want to stay in Python while targeting D3, the d3blocks library lets you create interactive D3 charts with only a few lines of code. In 2025, D3 often provides more flexibility and a more web-native rendering than Plotly for custom visualizations.

This is a chart title

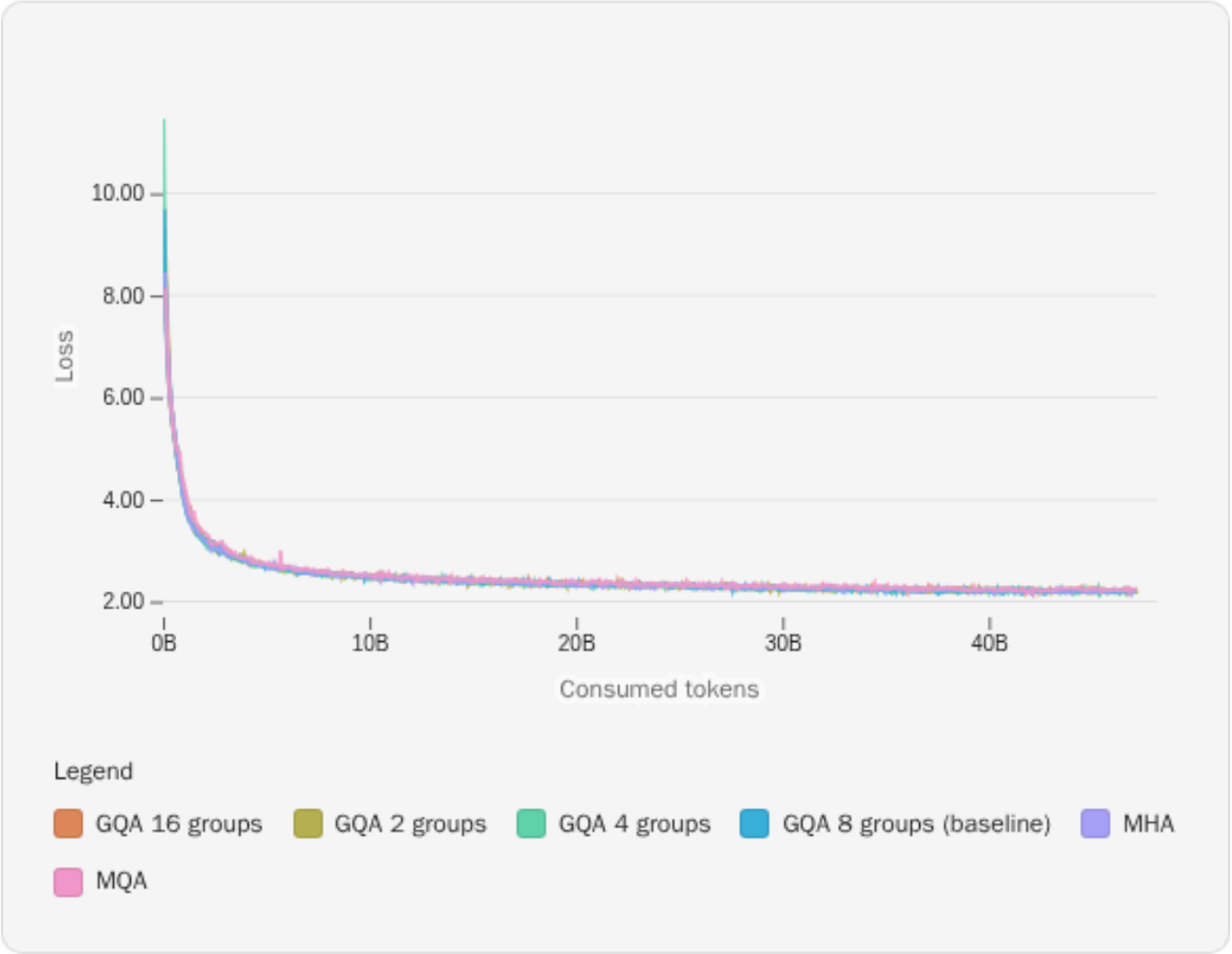


Figure 6 · Figure X: Some chart description

Credit: Example

Prop	Required	Description
`src`	Yes	Path to the embed file in the `embeds` folder.
`title`	No	Short title displayed above the card.
`desc`	No	Short description displayed below the card. Supports inline HTML (e.g., links).
`frameless`	No	Removes the card background and border for seamless embeds.
`align`	No	Aligns the title/description text. One of `left` (default), `center`, `right`.
`id`	No	Adds an `id` to the outer figure for deep-linking and cross-references.
`data`	No	Path (string) or array of paths (string[]) to data file(s) consumed by the embed.
`config`	No	Optional object for embed options (e.g., `{ defaultMetric: 'average_rank' }`).

Code example

DATA

If you need to link your HTML embeds to data files, there is an `assets/data` folder for this. As long as your files are there, they will be served from the `public/data` folder. You can fetch them with this address: `[domain]/data/your-data.ext`

> ⚠️ Be careful, unlike images, data files are not optimized by Astro. You need to optimize them manually.

Dataviz page

The template includes a dedicated `/dataviz` page that automatically collects and displays all visual assets from your article in one place. It's useful for reviewing your figures, exporting them, or sharing a visual summary of your paper.

What it extracts:

Type	Source	Details
Charts	`<HtmlEmbed>` components	Rendered live with all props (config, data, wide)
Images	`<Image>` components	Resolved from MDX imports to actual files
Stacks	`<Stack>` containing `<Image>`	Grouped as a single item, preserving layout
Tables	Markdown tables (`| ... |`)	Parsed with headers and rows

The page is accessible at `/dataviz` (see it live) and displays each item with:

- A type badge (Chart, Image, Stack, Table)
- A sequential number (#1, #2, ⋯)
- The title or caption when available
- A "View in article" link back to the source

Excluding items from the gallery:

Add the `skipGallery` prop to any `<Image>` or `<HtmlEmbed>` you don't want on the dataviz page. For `<Stack>` components, if all images inside have `skipGallery`, the entire stack is excluded.

<details>
<summary>Code example</summary>
</details>

Downloading charts:

Each chart on the dataviz page has a download button (PNG export). You can also batch-export all charts as a ZIP by running this in the browser console:

```
window.exportAllEmbeds()
```

Vibe coding charts

> ⚠️ This is a work in progress. It may change quickly.

Prompt

This page explains how to use the directives to author D3 charts as self-contained HTML fragments. Using claude code works better.

The goal is to make responsive, accessible, interactive and dark mode ready charts.

1. The embed directives are available as a Cursor skill in `.cursor/skills/create-html-embed/` (the agent picks it up automatically). Full conventions: `.cursor/skills/create-html-embed/directives.md`.
2. Opt: use an already existing chart as a starting point.
3. Ask claude to code the chart. Here's a typical prompt:

```
I want you to code a new d3 chart named `yourchart`.
I have one CSV file called `yourdata.csv` in the data folder.
The csv has the following columns: `x`, `y`, `z`.
I want you to code a d3 chart that visualizes the data.
```

4. Once the chart created, iterate with littles adjustments to make it better.
5. And that's it! 🎉

Base examples

These are fundamental chart types that serve as building blocks for more complex visualizations. You can find all the source code in `app/src/content/embeds`.

Post-Training Adventure

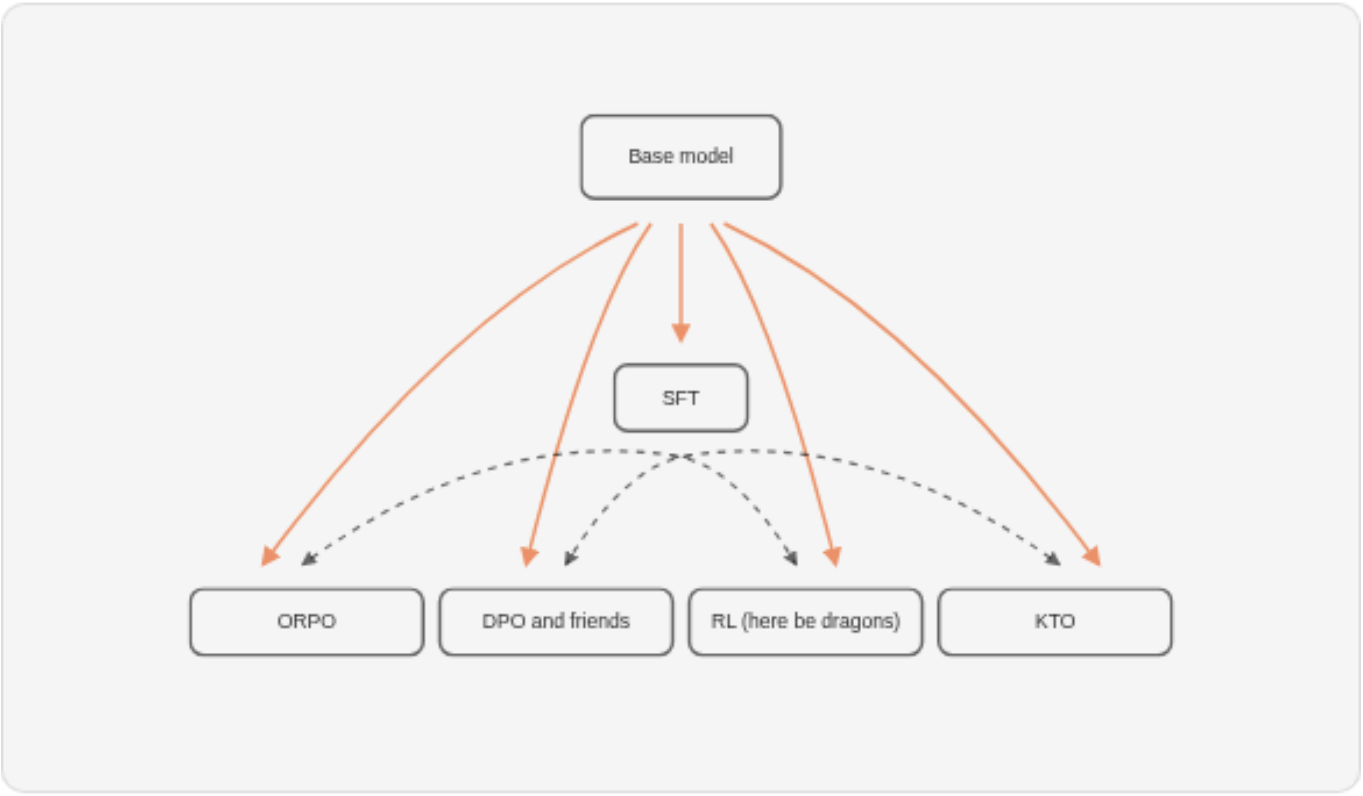


Figure 7 · Interactive visualization of a model's post-training journey, showing the different stages and decisions.

Advanced examples

These are more complex, interactive visualizations that demonstrate advanced D3 capabilities and real-world applications.

Embed not found

The requested embed could not be loaded: `d3-hf-citations.html`

The Experiment Map — Source × Prompt × Model

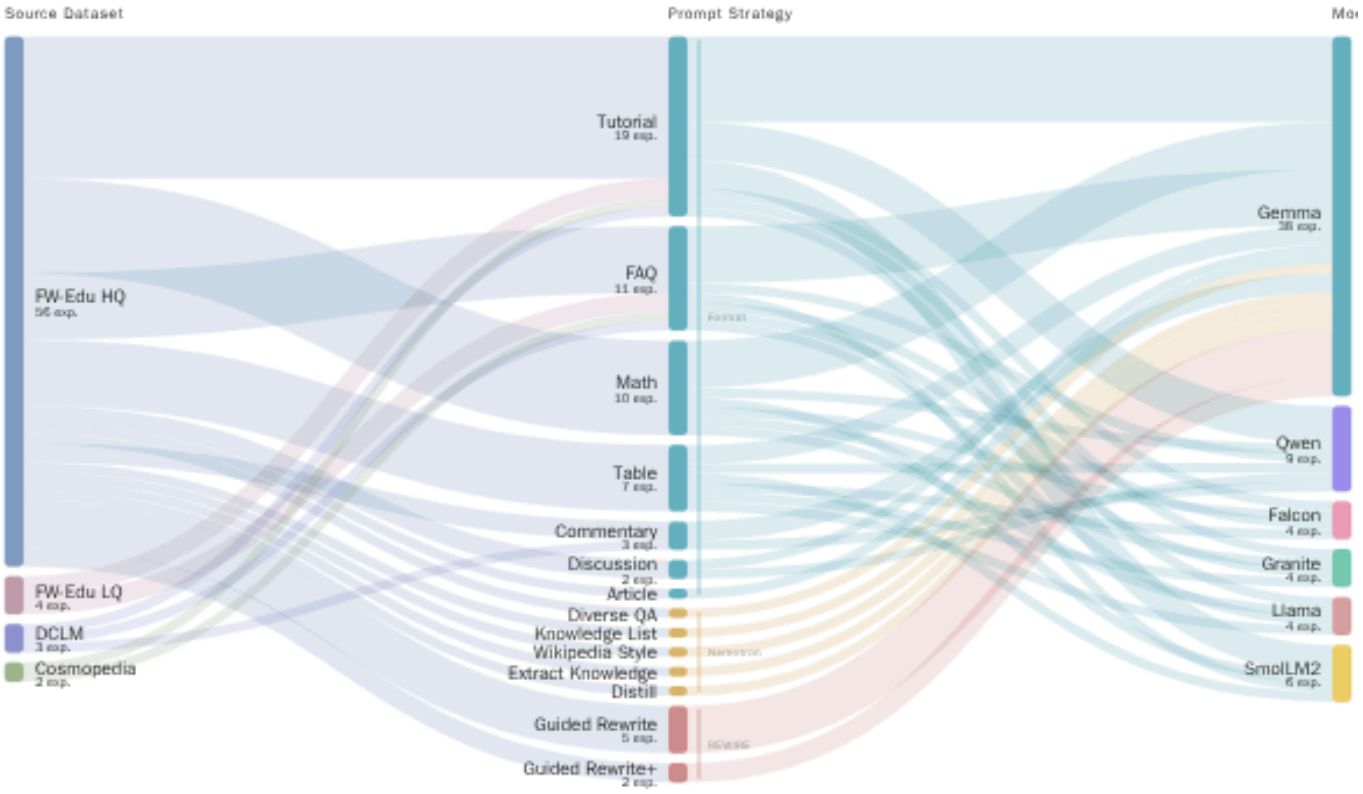


Figure 9 · Sankey diagram showing the flow of 65 experiments through source dataset, prompt strategy, and model family. Width = number of experiments.

Parameter Calculator

Vocabulary Size 128 k

Hidden Size 2048

Layers 16

Attention Heads 32

KV Heads 32

Intermediate Size 8192

Tie Embeddings

Yes

1.34 B

Parameters

EMBEDDINGS

Input + Output Projection

262 M

`128 k × 2048`

vocab_size × hidden_size

ATTENTION LAYERS

Q, K, V, O projections

268 M

`16 × 2048² × 4`

layers × hidden_size² × 4

FEED FORWARD

Up, Gate, Down projections

805 M

`16 × 2048 × 8192 × 3`

layers × hidden_size × intermediate_size × 3

LAYER NORMS

Input + Attention norms

68 K

`16 × 2048 × 2`

layers × hidden_size × 2

ATTENTION TYPE	EMBEDDING STRATEGY	PARAMS PER LAYER
MHA	Tied	67 M

Figure 10 · Interactive parameter calculator to visualize the parameter distribution of a dense transformer model. Useful for making architectural decisions or configuring ablations.

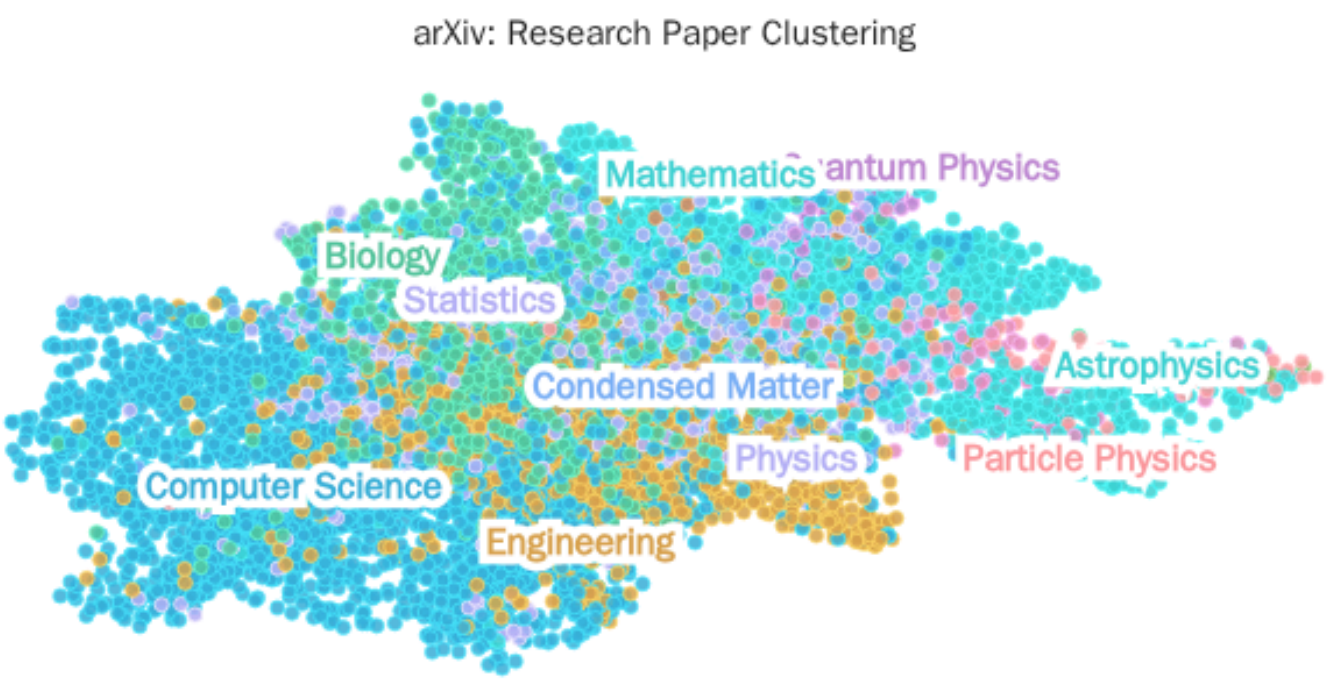


Figure 11 · Interactive visualization of ~8k recent arXiv submissions via UMAP dimensionality reduction. Each point represents a research paper positioned by semantic similarity. Colors indicate academic categories (cs.AI, cs.LG, cs.CV, etc.).

Figure 12 · Interactive 2D visualization of 382 Google Fonts clustered by visual similarity via UMAP. Each point represents a typeface positioned based on pixel-level differences computed from font matrices.

MNIST-like Neural Network

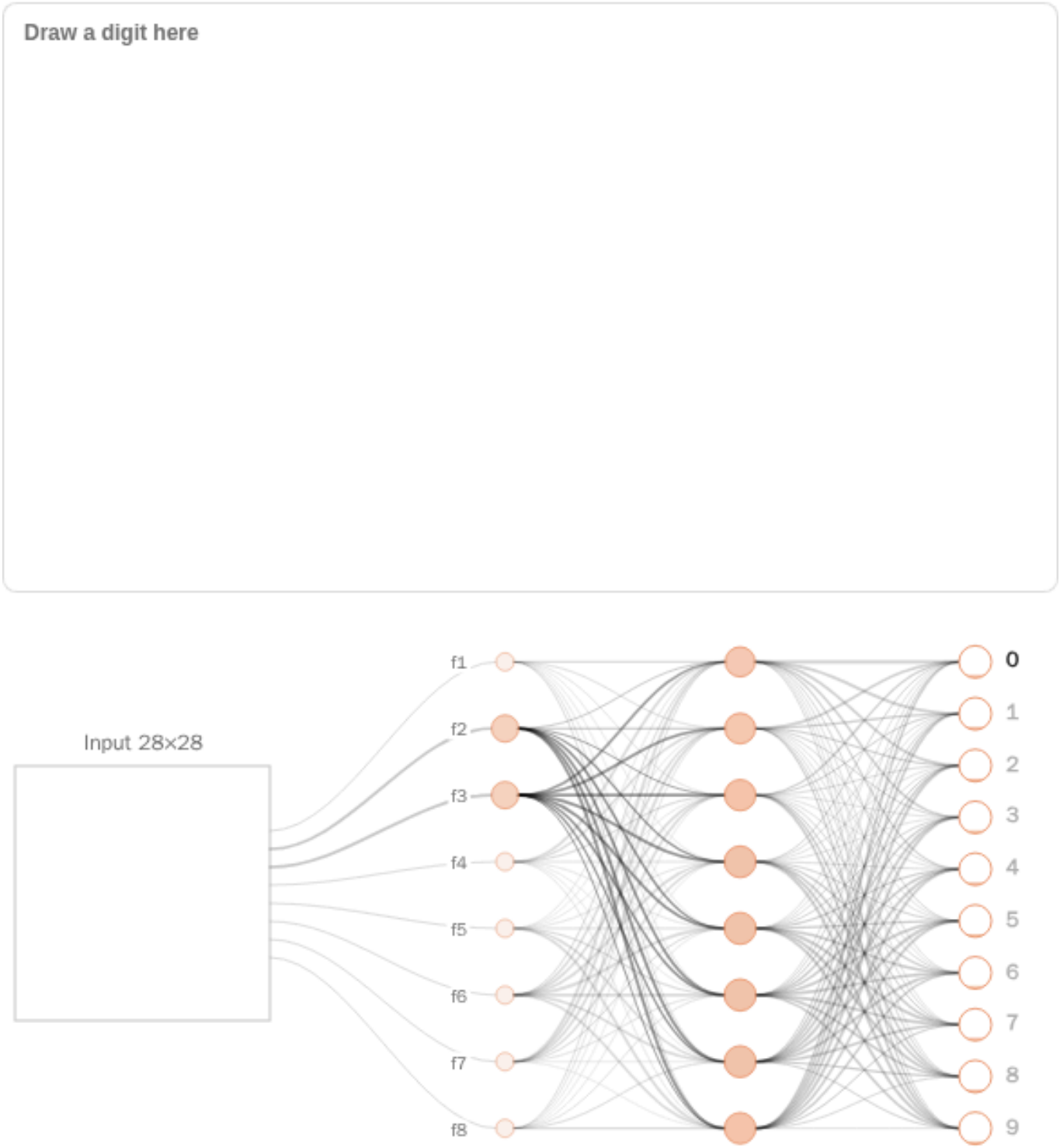


Figure 13 · Interactive MNIST-like neural network. Draw a digit on the left canvas; activations propagate through hidden layers (node size/opacity reflect activation). The right side displays class probabilities (0–9) with the top class emphasized.

Parameter Comparison

Choose a Model

Llama 3.2 1B | SmolLM3 3B | Llama 3.1 8B | Llama 3.1 70B

Export JSON | View Original

1.24 B

Parameters

LAYERS	HIDDEN SIZE	HEADS (Q/KV)	INTERMEDIATE	EMBEDDINGS
16	2048	32/8	8192	Tied

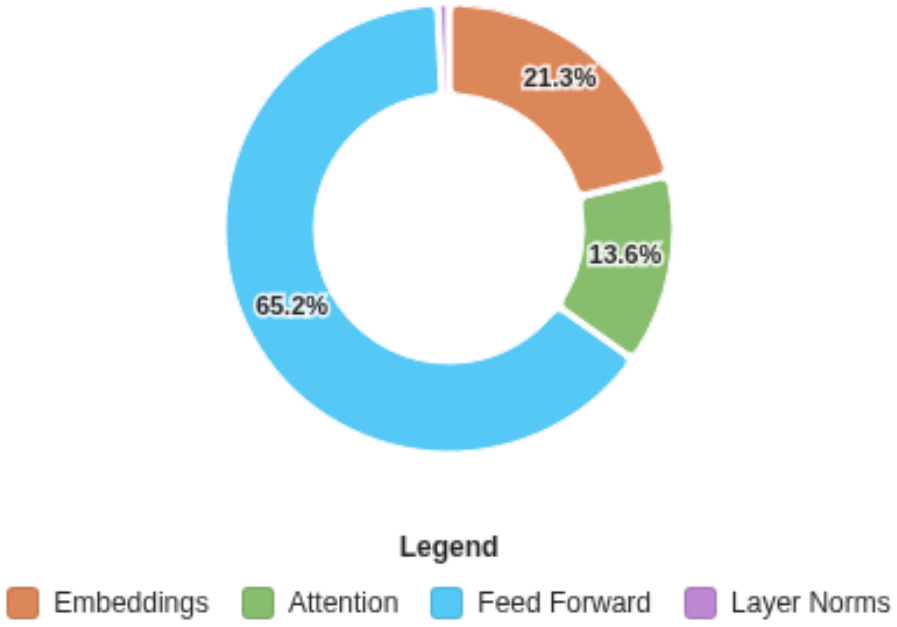


Figure 14 · Visual comparison of parameter distribution between different model sizes (8B vs 70B). Shows how embeddings represent a larger fraction in smaller models.

Attention Mechanisms

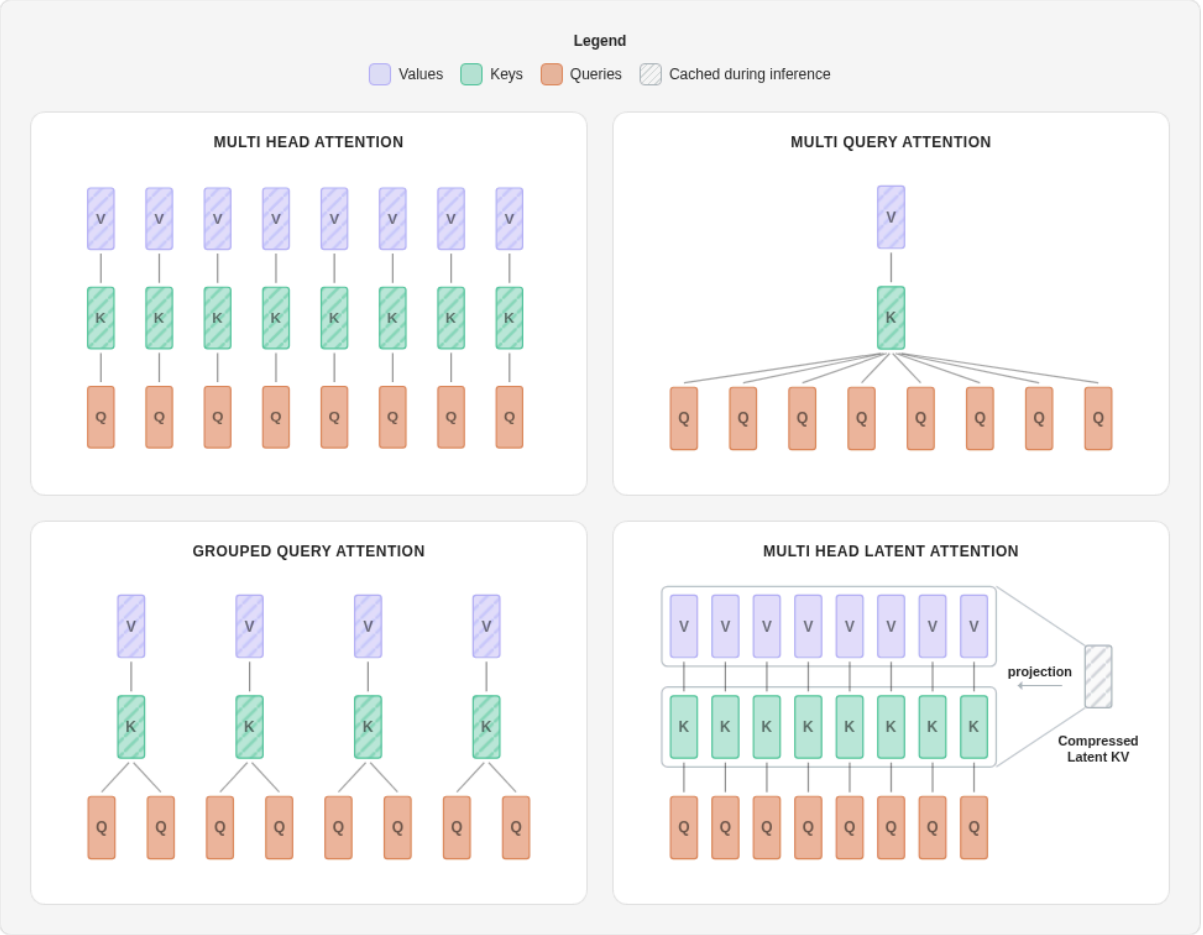


Figure 15 · Simplified illustration of Multi-Head Attention (MHA), Grouped-Query Attention (GQA), Multi-Query Attention (MQA), and Multi-head Latent Attention (MLA). Demonstrates how MLA compresses keys and values into a latent vector.

AWS Bandwidth Bottleneck

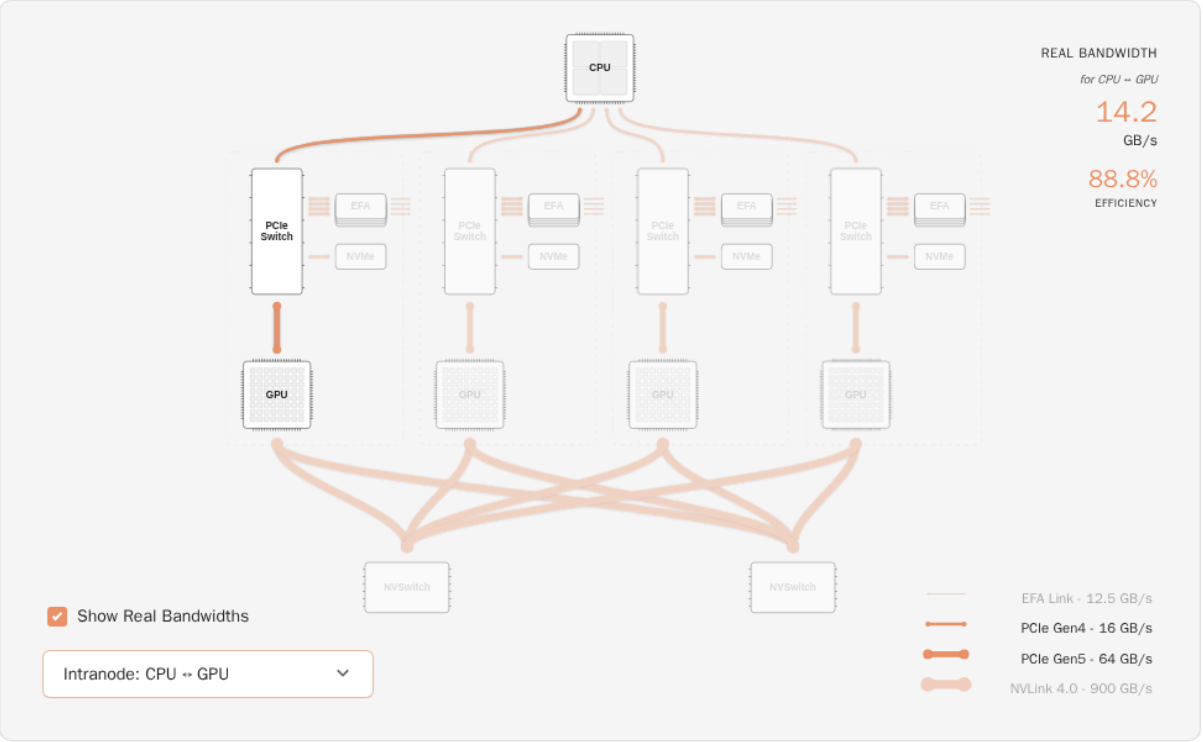


Figure 16 · Interactive diagram of key components and communication links in an AWS p5 instance. Shows the different GPU-GPU communication paths with their actual bandwidths.

Storage Performance Heatmaps

Figure 17 · Interactive heatmaps comparing storage system performance (scratch, root, FSx, admin) across different thread counts and I/O sizes. Visualizes throughput (GiB/s) and IOPS to identify optimal configurations for each storage tier.

Should You Train Your Own Model?

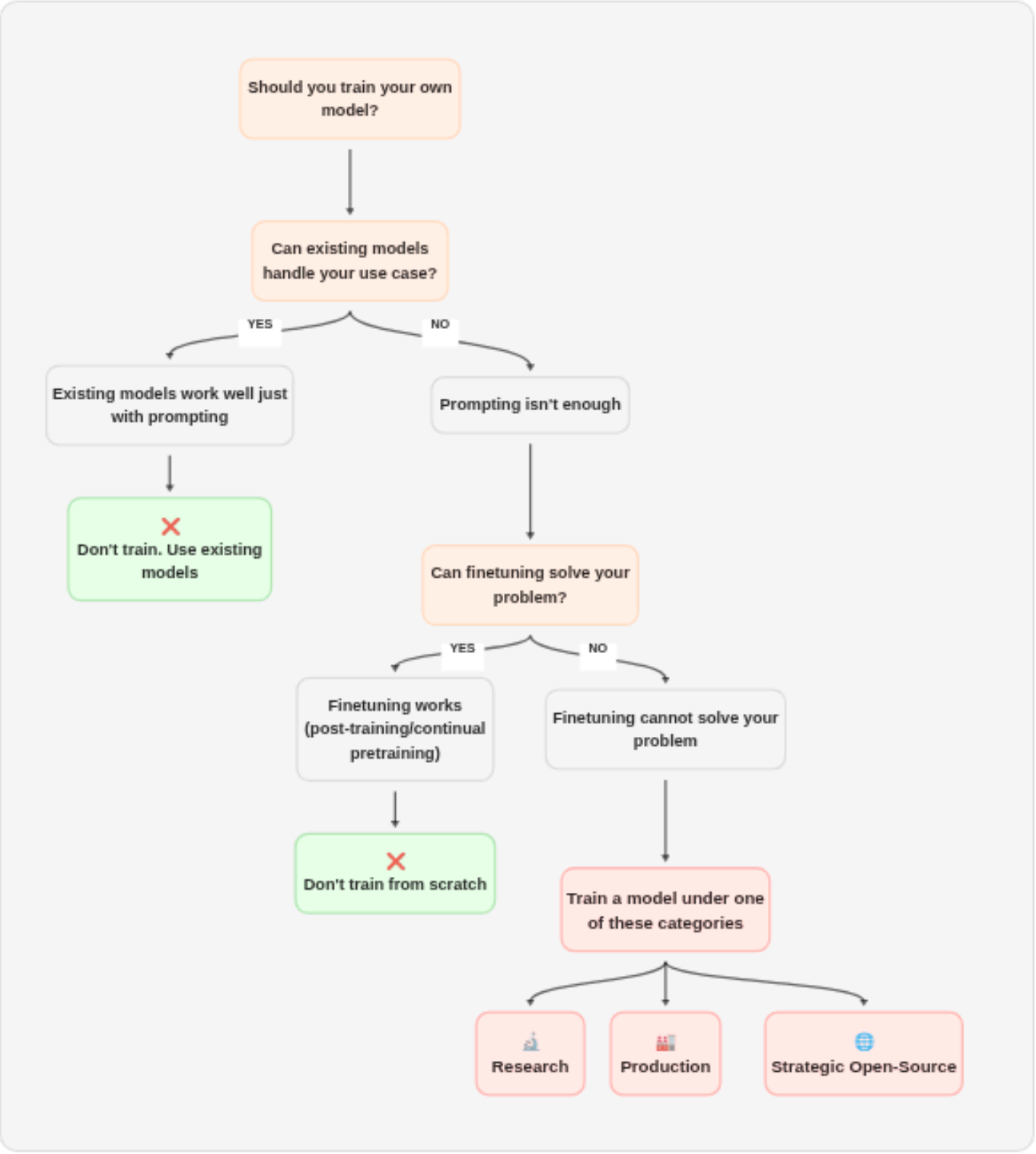


Figure 18 · Interactive decision tree to determine whether you should train your own model. Guides the thought process through key questions: Can existing models handle your use case? Should you fine-tune or train from scratch? For research, production, or strategic open-source?

Generic Line Chart

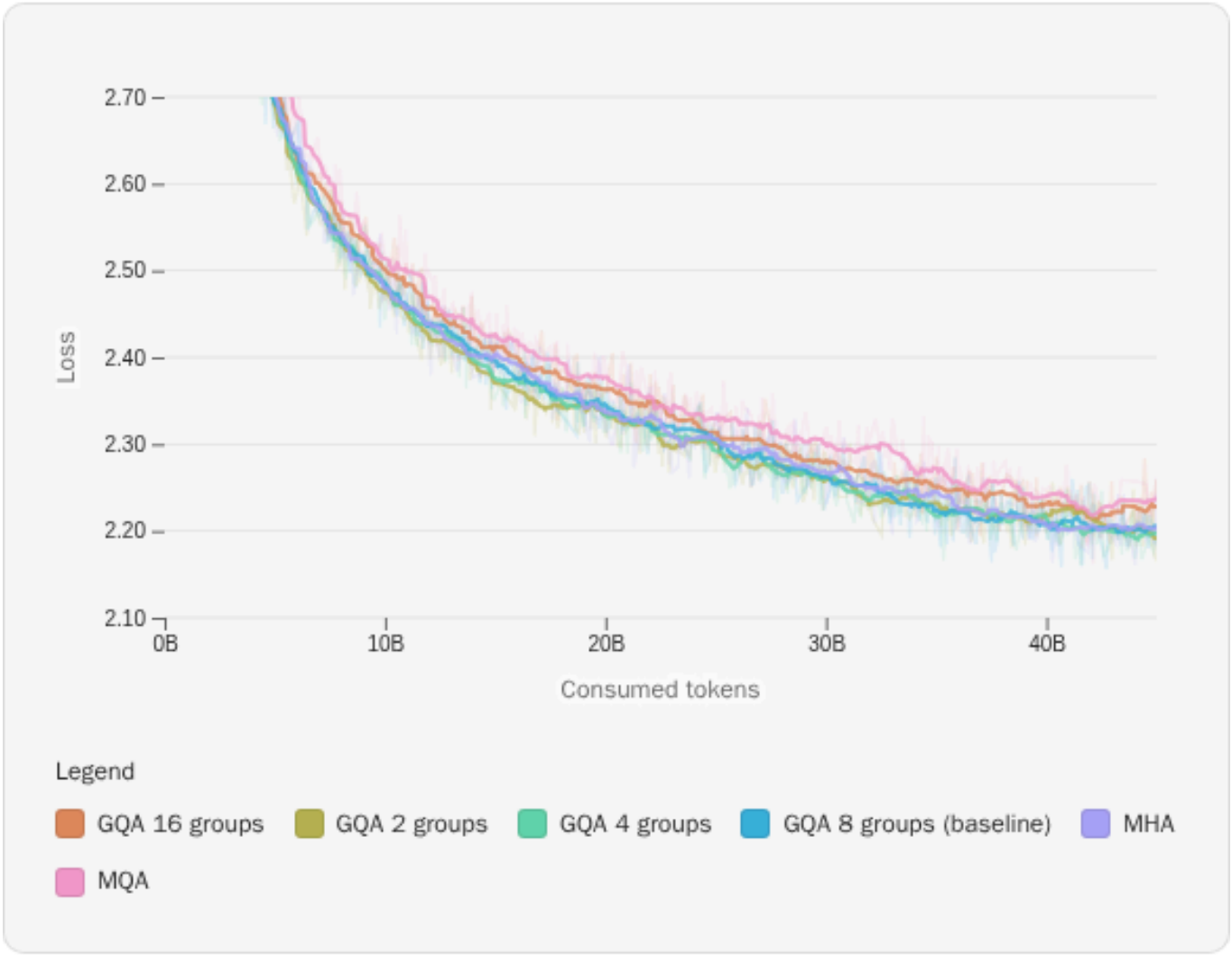


Figure 19 · Configurable line chart with zoom/pan, smoothing and interactive tooltips. Example with loss data for different attention configurations.
Credit: SmolLM Training Playbook

Generic Six Line Charts Grid

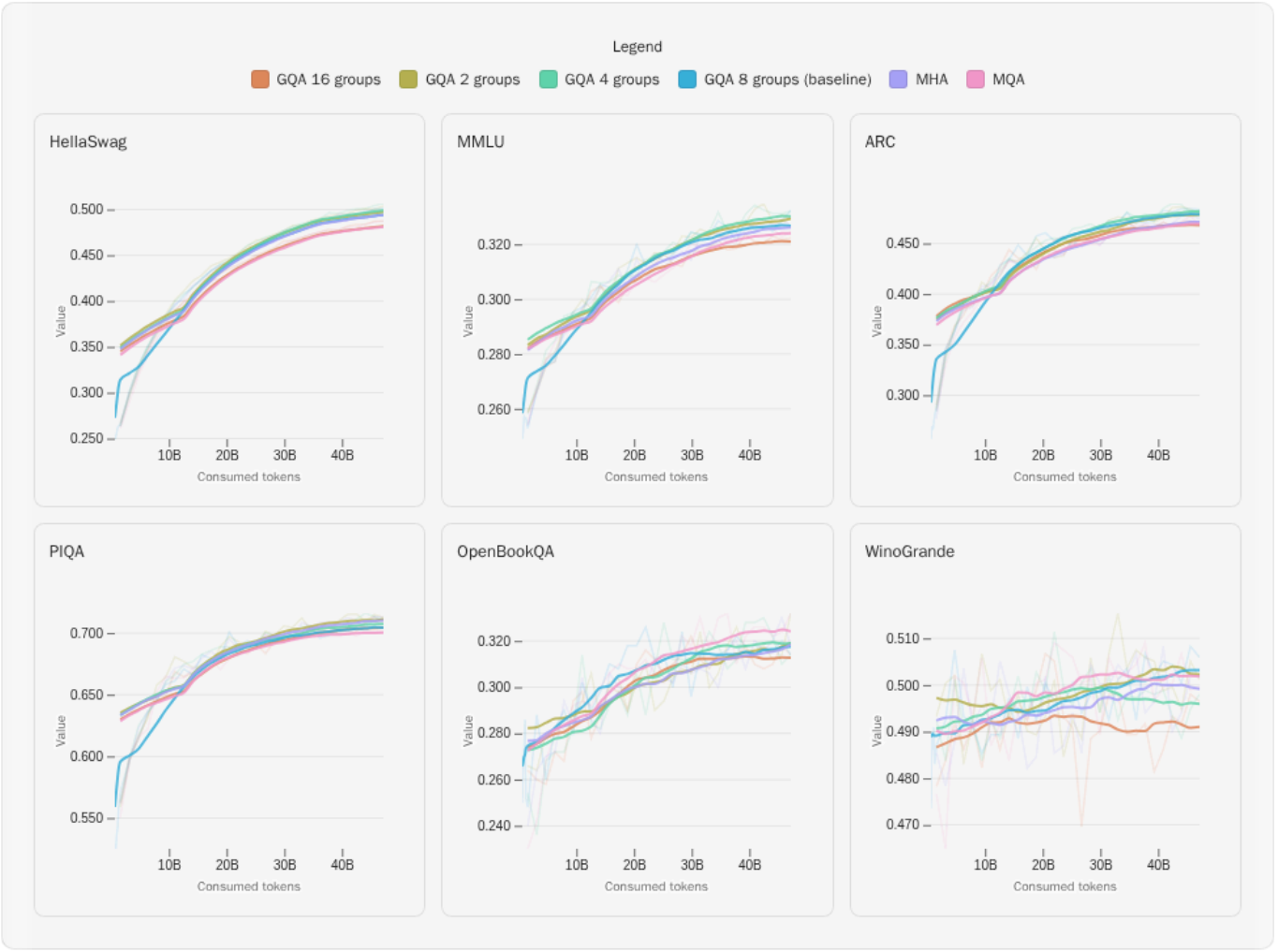
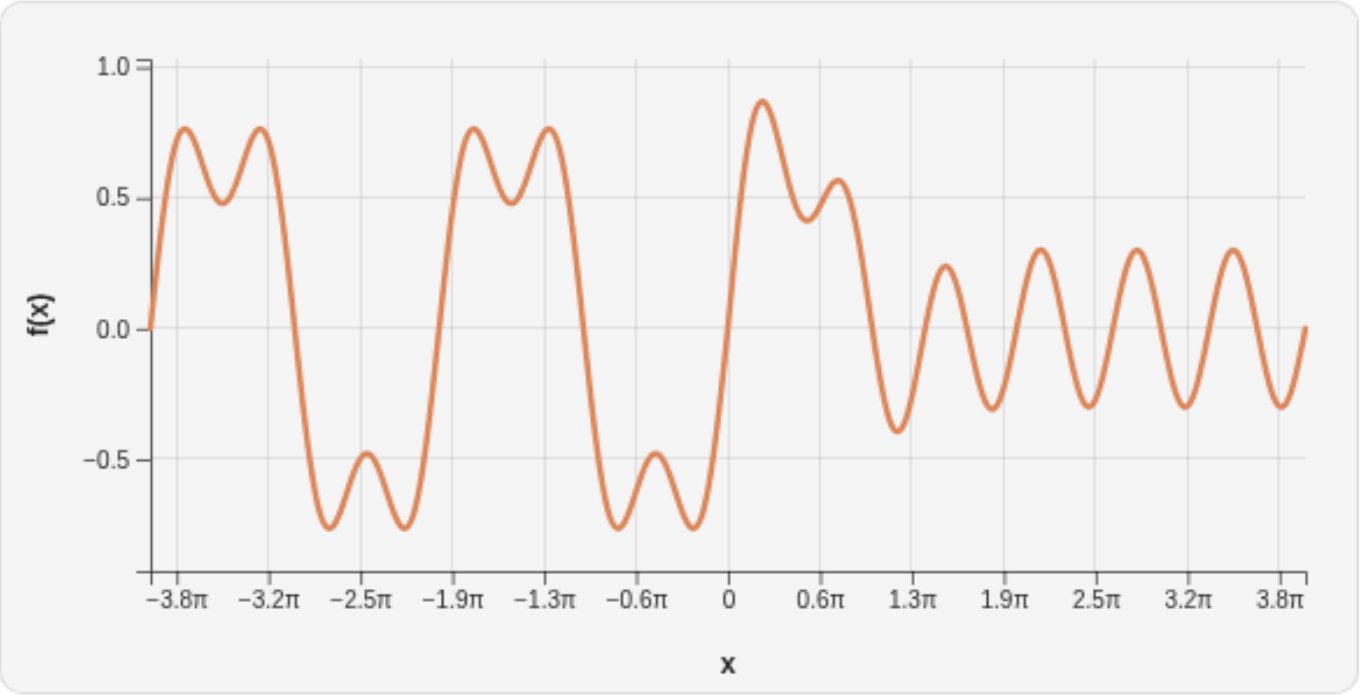


Figure 20 · Configurable grid of line charts with synchronized zoom/pan. Example with evaluations of different metrics.

Credit: SmolLM Training Playbook

Interactive Mathematical Function Plotter



Equation f(x) =

```
sin(x) * exp(-x^2/8) + 0.3*sin(3*
```

Domain

[-4π, 4π]

sin(x) * exp(-x^2/8) | sin(x) + 0.5*cos(2*x) | x^3 - 3*x

sin(x) * exp(-x^2/8) + 0.3*sin(3*x) | exp(-x^2/2) * cos(4*x)

sin(x) + sin(3*x)/3 + sin(5*x)/5

Figure 21 · Interactive equation editor with real-time function plotting. Edit mathematical expressions and see their graphs update instantly. Supports common functions (sin, cos, exp, etc.) with customizable domain range.

Import from LaTeX

> ⚠️ Experimental — May not work with all LaTeX documents.

Transform LaTeX papers into interactive web articles.

Quick Start

```
cd app/scripts/latex-importer/
cp your-paper.tex input/main.tex
cp your-paper.bib input/main.bib
node index.mjs
```

What Gets Converted

- `\label{eq:name}` → Interactive equations
- `\ref{eq:name}` → Clickable links
- `\includegraphics{}` → `<Image>` components
- Bibliography integration

Prerequisites

- Pandoc (`brew install pandoc`)
- LaTeX source files and figures

Docker Deployment

Set `ENABLE_LATEX_CONVERSION=true` in your Hugging Face Space to enable automatic conversion during build.

Import from Notion

> ⚠️ Experimental — May not work with all Notion pages.

Convert Notion pages into interactive web articles.

Prerequisites

You need 2 things:

1. NOTION_TOKEN — Your Notion integration token
   - Go to notion.so/my-integrations
   - Create a new integration → copy the token (starts with `secret_` or `ntn_`)
2. NOTION_PAGE_ID — The ID of your Notion page

- Open your page in browser: `https://www.notion.so/My-Page-abc123def456`
- The ID is the last part after the title: `abc123def456`

Share Your Page with the Integration

> This step is required! Having a token is not enough — you must explicitly share the page with your integration.

1. Open your Notion page
2. Click "⋯" (top right corner)
3. Click "Connections" or "Add connections"
4. Select your integration from the list
5. Confirm access

If your page is nested inside another page or database, you can either share the parent page (the integration will have access to all sub-pages) or share each page individually.

Option A: Automatic on HF Space

Add these to your Space settings (Settings → Variables and secrets):

Type	Name	Value
Secret	`NOTION_TOKEN`	your token
Variable	`NOTION_PAGE_ID`	your page ID
Variable	`ENABLE_NOTION_CONVERSION`	`true`

Then restart the Space. Every rebuild will fetch the latest Notion content automatically.

Option B: Local Development

```
cd app/scripts/notion-importer/
npm install
cp env.example .env
# Edit .env with NOTION_TOKEN and NOTION_PAGE_ID
npm run notion:import   # or: node index.mjs
```

This rewrites `src/content/article.mdx` with your Notion content. Push to deploy.

What Gets Converted

- Images (downloaded locally)

- Callouts → `<Note>` components
- Enhanced tables and code blocks
- Smart link conversion

Writing Tips

Simple guidelines to make your research writing clear, engaging, and effective. Focus on what matters most: helping readers understand your ideas quickly and completely.

Short sections

Break content into small, purpose-driven sections. Each section should answer a single question or support one idea. This improves scanability, helps readers navigate with the TOC, and makes later edits safer.

Clear, minimal annotations

Favor concise captions and callouts that clarify what to look at and why it matters. In code, highlight just the lines that carry the idea; avoid verbose commentary. Precision beats volume.

Explain math notation

Introduce symbols and variables the first time they appear, and prefer well-known identities over custom shorthand. When formulas carry the message, add one sentence of plain-language interpretation right after.

> For example, in linear regression with features $x \in \mathbb{R}^d$, weights $w \in \mathbb{R}^d$, and bias b, the prediction is:
>
> $$\hat{y} = w^\top x + b$$
>
> A common training objective is the mean squared error over N samples:
>
> $$\mathcal{L}(w, b) = \frac{1}{N} \sum_{i=1}^{N} (w^\top x_i + b - y_i)^2$$
>
> Interpretation: the model fits a hyperplane that minimizes the average squared prediction error.

Use the right chart

Picking the right visualization depends on your goal (compare values, show distribution, part-to-whole, trends, relationships, etc.). The Visual Vocabulary poster below provides a concise mapping from analytical task to chart types.

Figure 22 · Credits Financial-Times
A handy reference to select chart types by purpose — click to enlarge.

Greetings

Huge thanks to the following people for their precious feedbacks!

Leandro von Werra
@lvwerra

Clémentine Fourrier
@clefourrier

Hynek Kydlíček
@hynky

Luis Wiedmann
@lusxvr

Pablo Montalvo-Leroux
@molbap

Lewis Tunstall
@lewtun

Guilherme Penedo
@guipenedo

Francesco Capuano
@fracapuano

Citation

For attribution in academic contexts, please cite this work as

```
Thibaud Frere (2025). "Bringing paper to life: A modern template for scientific writing".
```

BibTeX citation

```
@misc{frere2025_bringing_paper_to_life_a_modern_template_for_scientific_writing,
  title={Bringing paper to life: A modern template for scientific writing},
  author={Thibaud Frere},
  year={2025},
  doi={10.1234/abcd.efgh}
}
```

DOI

10.1234/abcd.efgh

References

1. Cover, T. M., & Thomas, J. A. (2006). Entropy, Relative Entropy, and Mutual Information. In *Elements of Information Theory* (2nd ed., pp. 13–55). Wiley. ↑

2. Doe, J. (2020). *Learning Efficient Representations for Large-Scale Visual Recognition* [Phdthesis, Massachusetts Institute of Technology]. 10.5555/mit-2020-xyz ↑

3. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. 10.1109/CVPR.2016.90 ↑ back: 1, 2

4. Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1412.6980 ↑ back: 1, 2, 3

5. McKinney, W. (2017). *Python for Data Analysis* (2nd ed.). O’Reilly Media. ↑ back: 1, 2

6. OpenAI. (2023). *GPT-4 Technical Report* (Techreport arXiv:2303.08774). OpenAI. https://arxiv.org/abs/2303.08774 ↑

7. Pedregosa, F., Varoquaux, G., Gramfort, A., & others. (2024). *scikit-learn: Machine Learning in Python (Version 1.4)*. Software. 10.5281/zenodo.592264 ↑

8. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., & others. (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. arXiv preprint. 10.48550/arXiv.1910.10683 ↑

9. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., & others. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*(7676), 354–359. 10.1038/nature24270 ↑

10. Smith, E., Zhang, W., Rossi, M., & Patel, N. (2024). Privacy-Preserving Training with Low-Precision Secure Aggregation. In A. Smith & B. Johnson (Eds.), *Proceedings of the 41st International Conference on Machine Learning* (Vol. 235, pp. 12345–12367). PMLR. https://proceedings.mlr.press/v235/ ↑

11. Smith, J., Lee, A., & Kumar, R. (2021). *ImageNet-21K Subset (Version 2.0)*. Dataset on Zenodo. 10.5281/zenodo.1234567 ↑

12. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*. ↑ back: 1, 2, 3

Footnotes

1. Footnote attached to the sentence above. ↑ back: 1, 2

2. Multi-paragraph footnote. First paragraph.

   Second paragraph with a link to Astro. ↑

3. Footnote with a list: ↑

   - First item
   - Second item

4. Footnote with code snippet:

```
function add(a: number, b: number) {
  return a + b;
}
```

Result: `add(2, 3) === 5`. ↑

5. Footnote containing citation (Vaswani et al., 2017) and (Kingma & Ba, 2015). ↑

6. This footnote includes inline math $E = mc^2$ and a display equation:

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

This is the Gaussian integral, a fundamental result in probability theory. ↑