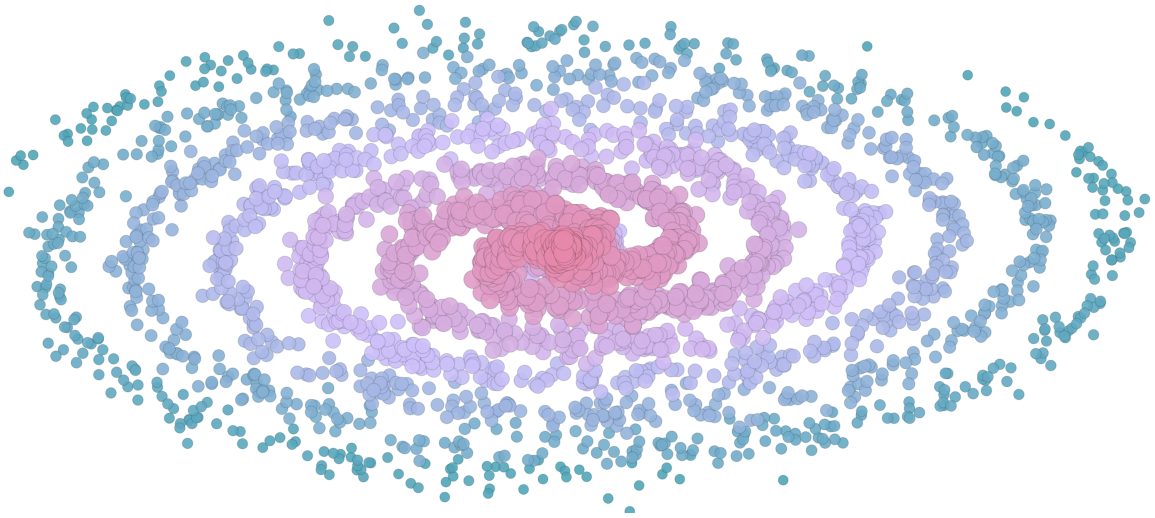
Bringing paper to life: A modern template for scientific writing



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

AUTHOR	AFFILIATION	PUBLISHED
Thibaud Frere	Hugging Face	Sep. 01, 2025

Welcome to this single-page 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.

FEATURES

- Markdown-based
- KaTeX math
- Syntax highlighting
- Citations in all flavors
- 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

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.

To give you a sense of what inspired this template, here is a short, curated list of well-designed and often interactive works from Distill:

- Growing Neural Cellular Automata
- Activation Atlas
- Handwriting with a Neural Network
- The Building Blocks of Interpretability

Getting Started

Installation

The recommended way is to duplicate this Space on Hugging Face rather than cloning it directly:

1. Open the Space: 🤗 science-blog-template
   and click `Duplicate this Space`.
2. Give it a name, choose visibility, and keep the free CPU instance.
3. Clone your new Space repository.

```
git clone git@hf.co:spaces/<your-username>/<your-space>
cd <your-space>
```

4. Use Node.js 20 or newer.
   To manage versions, consider using nvm

- macOS/Linux: see nvm-sh
- Windows: see nvm-windows

```
nvm install 20
nvm use 20
node -v
```

5. Install lfs and pull files from the repository.

```
git lfs install
git lfs pull
```

If you attempt to push binary files without Git LFS installed, you will encounter an error.

6. Install dependencies.

```
cd app
npm install
```

And that’s it!

You’re ready to go! 🎉

Development

```
npm run dev
```

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

Build

```
npm run build
```

Deploy

Every push automatically triggers a build and deploy on Spaces.

```
# Make edits locally, then:
git add .
git commit -m "Update content"
git push
```

> Serving the `dist/` directory on any static host is enough to deliver the site.

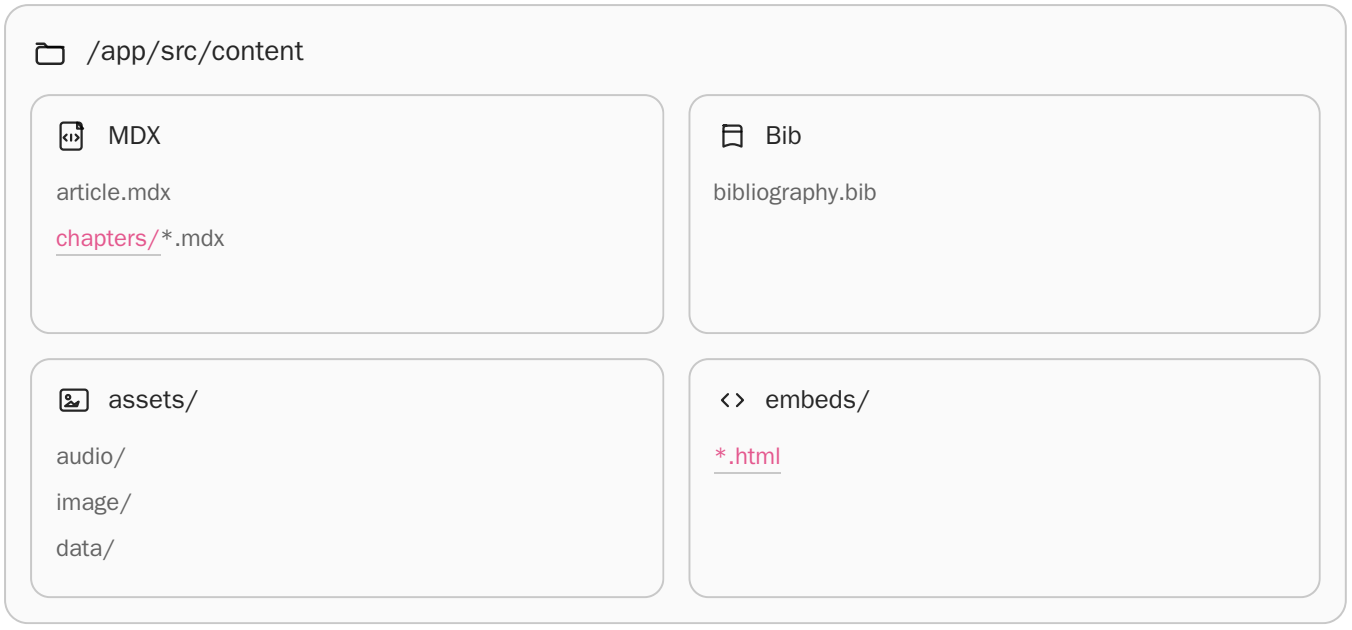
> A [slugified-title].pdf and thumb.jpg are also generated at build time.
> You can find them in the public folder and point to them at `[domain]/public/thumb.jpg`.

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
---
```

Frontmatter fields

CONTENT

Your story. Write your content here.

Content in app/src/content/article.mdx

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

<Sidenote>
  This paragraph is written in Markdown.
  <Fragment slot="aside">A short callout inserted via a component.
</Fragment>
  </Sidenote>
<ResponsiveImage 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

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

<MyChapter />
```

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

Hot Flamingoes
#E4588E
HSL 62, 72, 337°
RGB 228, 88, 142

HUE

337°

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. See also the complete Markdown documentation.

Math Code Citation Footnote Mermaid Separator Table Audio

Math

KaTeX is used for math rendering. You can use inline notation `$...$` or block `$$...$$` notation. As an example, this is an inline math equation: $x^2 + y^2 = z^2$ and this is a block:

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

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].

Code example

Reference

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

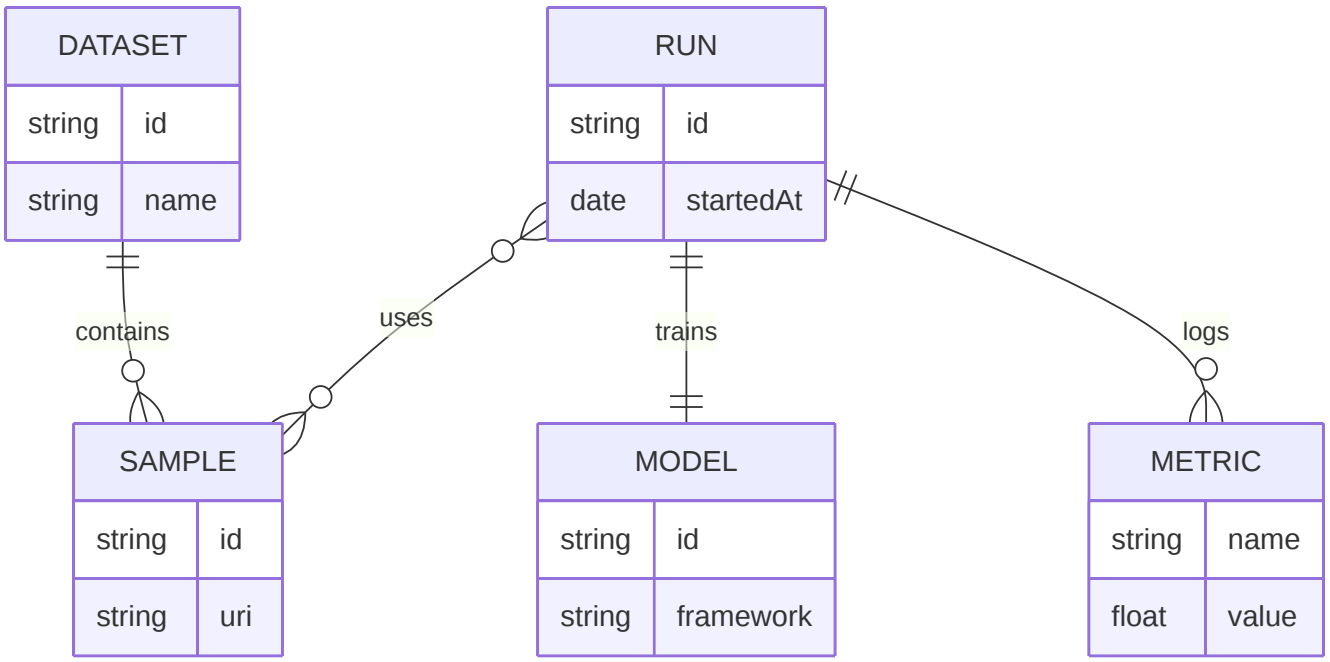
2. Image and chart
   You can make a link to an image or a chart by adding an ID on it.
   `<HtmlEmbed id="neural-network-mnist-like"/>` then you can link to it with a link like `<a href="#neural-network-mnist-like">Fig 1</a>`.

Example Chart 1 or Fig 1

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 `| --- |`.

Method	Score
A	0.78
B	0.86

Code example

Audio

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

0:00 / 16:57

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. You have to import them in the .mdx file you want to use them in.

ResponsiveImage Placement Accordion Note HtmlEmbed Iframe

ResponsiveImage

Responsive images automatically generate an 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.

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.

Code example

WIDE EXAMPLE

demo wide

Code example

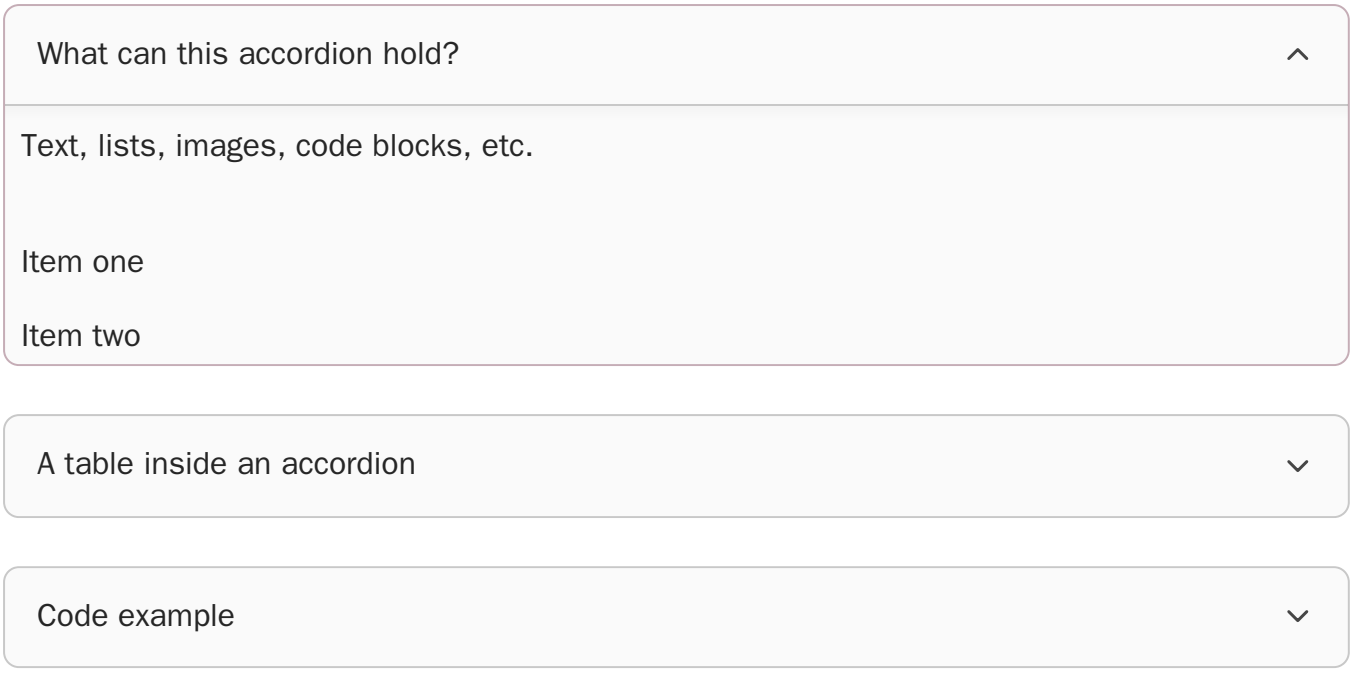
FULL-WIDTH EXAMPLE

demo full-width

Code example

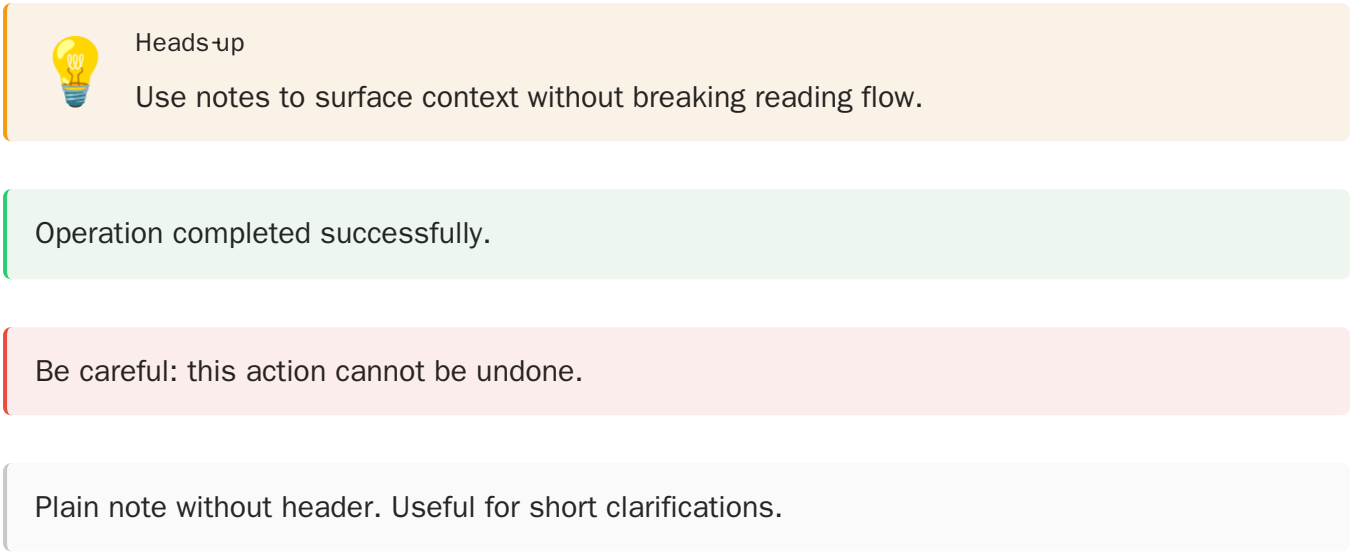
Accordion

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



Note

Small contextual callout for tips, caveats, or emphasis.



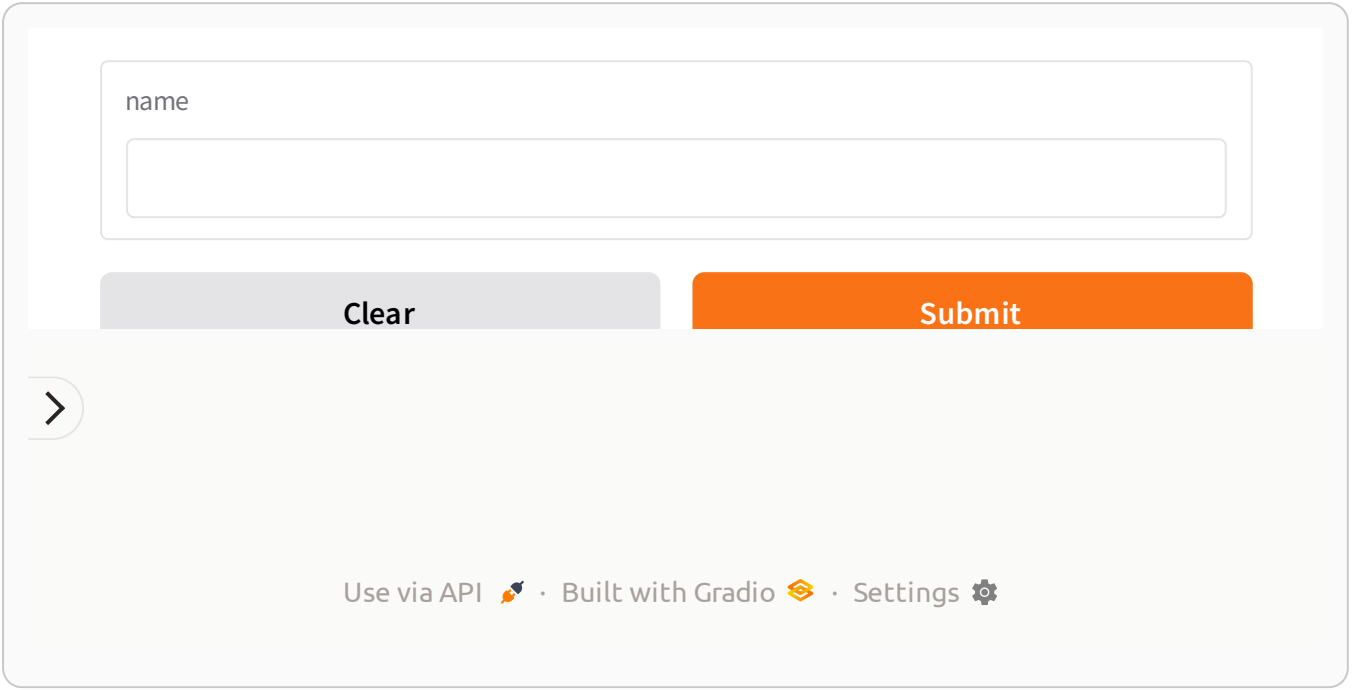
Prop	Required	Type	Description
`title`	No	string	Short title displayed in header
`emoji`	No	string	Emoji displayed before the title
`class`	No	string	Extra classes for custom styling
`variant`	No	'info'	'success'

Code example

Iframes

You can embed external content in your article using iframes. For example, TrackIO, 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

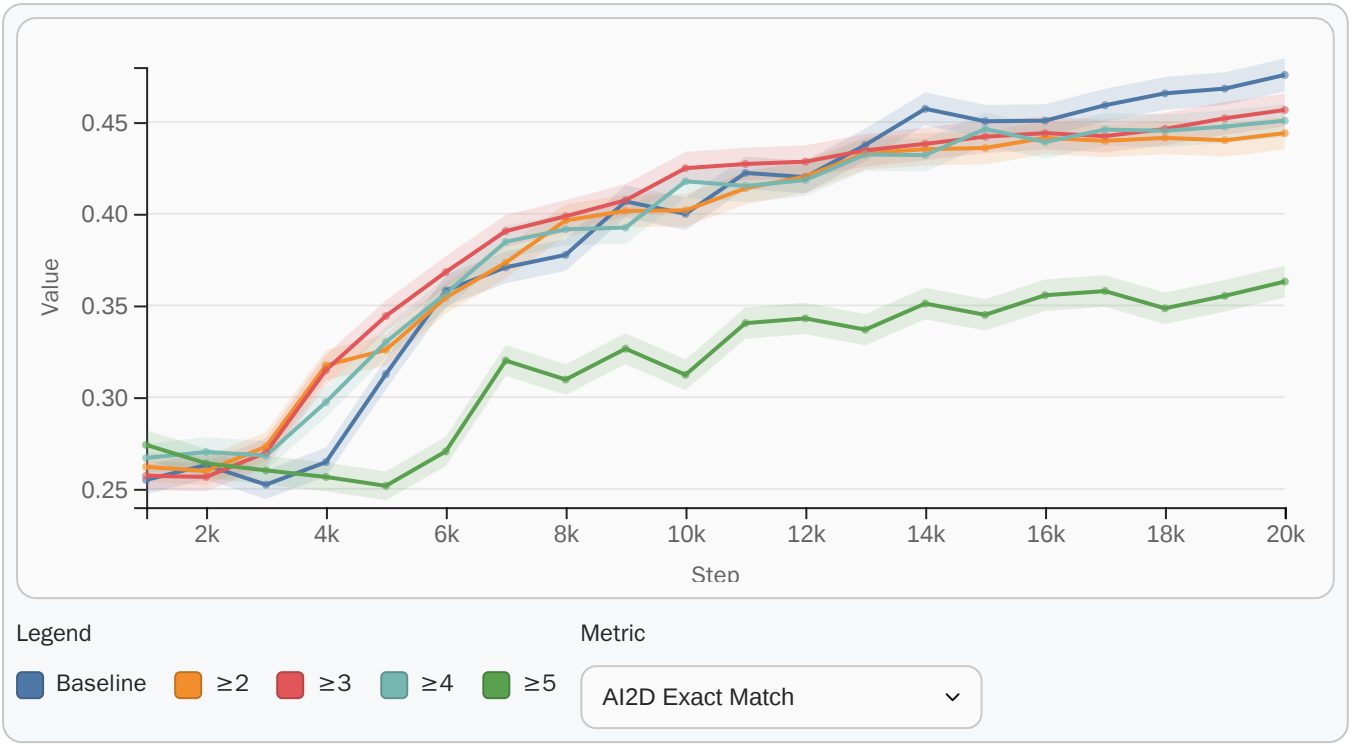


Figure X: Some chart description
Credit: Example

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., link
`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 emb
`config`	No	Optional object for embed options (e.g., `{ defaultMetric: 'average_r`

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.

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. Use this ref a a baseprompt: `app/src/content/embeds/vibe-code-d3-embeds-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! 🎉

Real-world examples

They can be found in the `app/src/content/embeds` folder and you can also use them as a starting point or examples to vibe code with.

d3-benchmark: LLM Benchmark

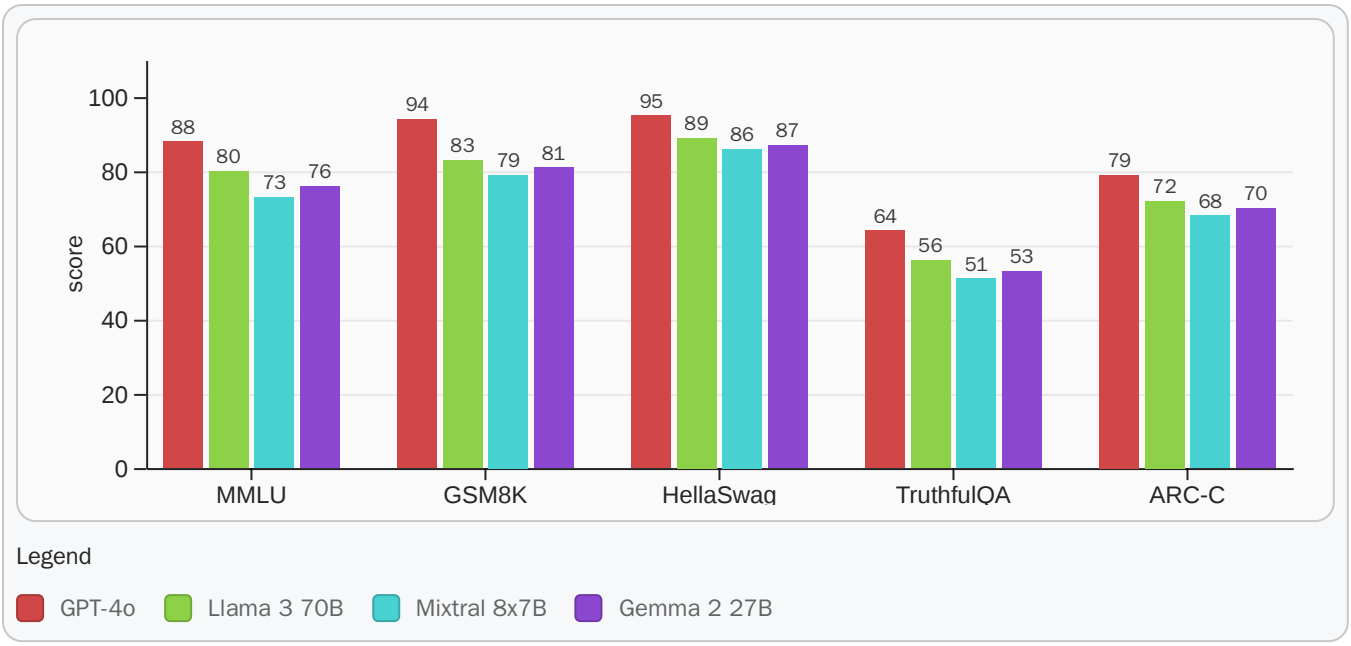


Figure 1: Grouped bar chart comparing model scores across benchmarks (MMLU, GSM8K, HellaSwag, TruthfulQA, ARC-C). Each group is a benchmark; colors encode models; values are accuracy/score (higher is better).

---

d3-line: Average Ranking of Models

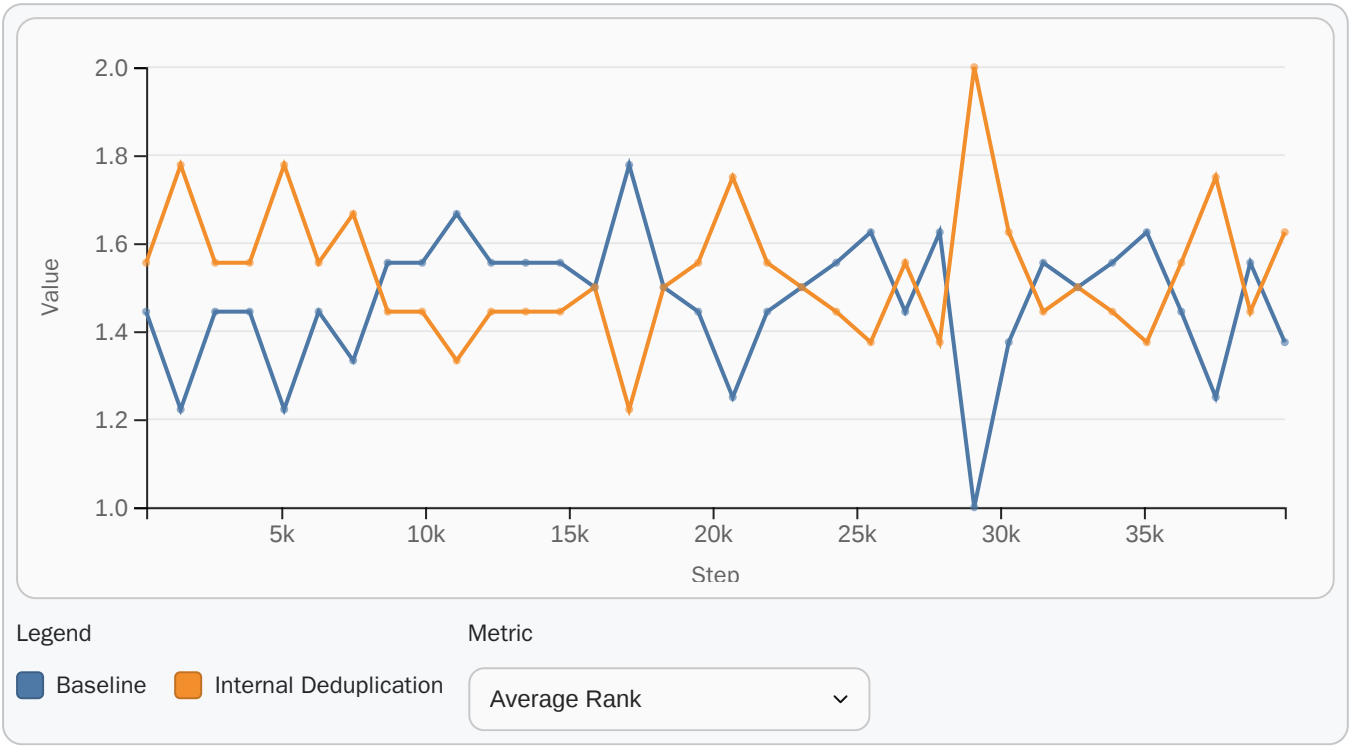


Figure 2: Average Ranking of Models trained with internally deduplicated / merged samples. No clear benefit in merging can be seen with respect to model performance.
Credit: FineVision

---

d3-neural-network: MNIST-like Neural Network

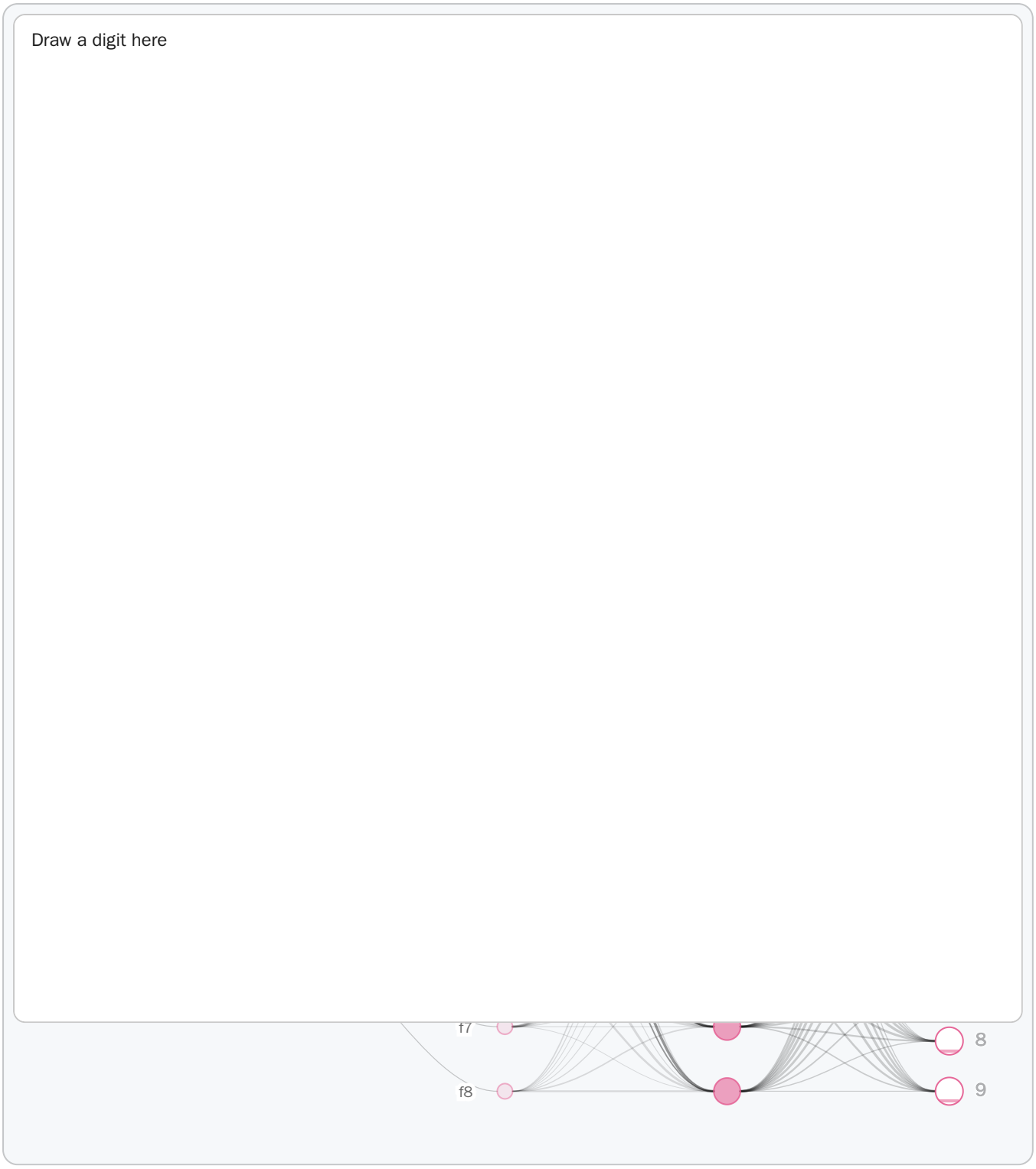


Figure 3: Interactive MNIST-like neural network. Draw a digit on the left canvas; activations propagate through hidden layers (node size/opacity reflect activation). The right side displays class probabilities (0–9) with the top class emphasized.

d3-matrix: Baseline and Δ (Improved − Baseline)

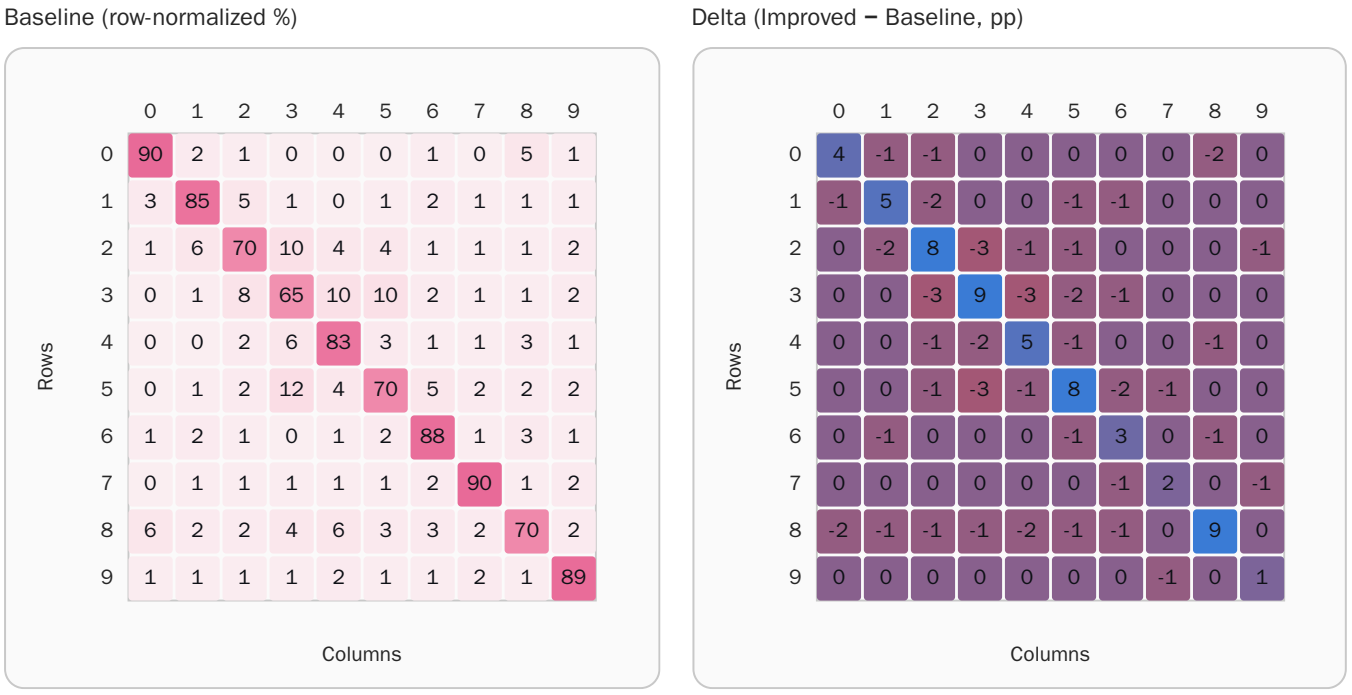
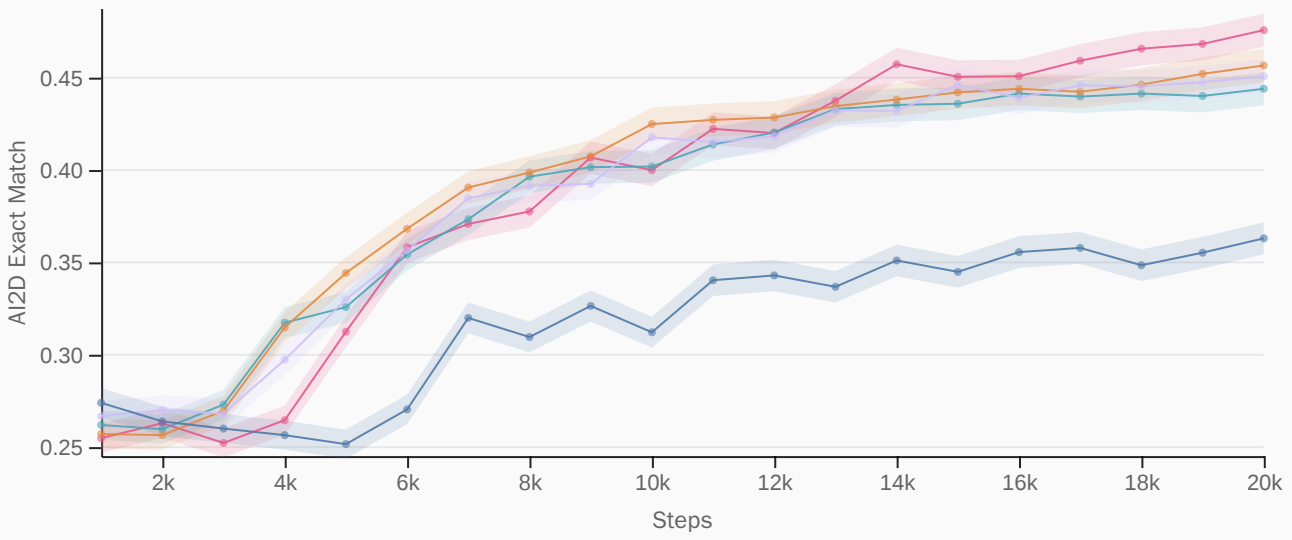


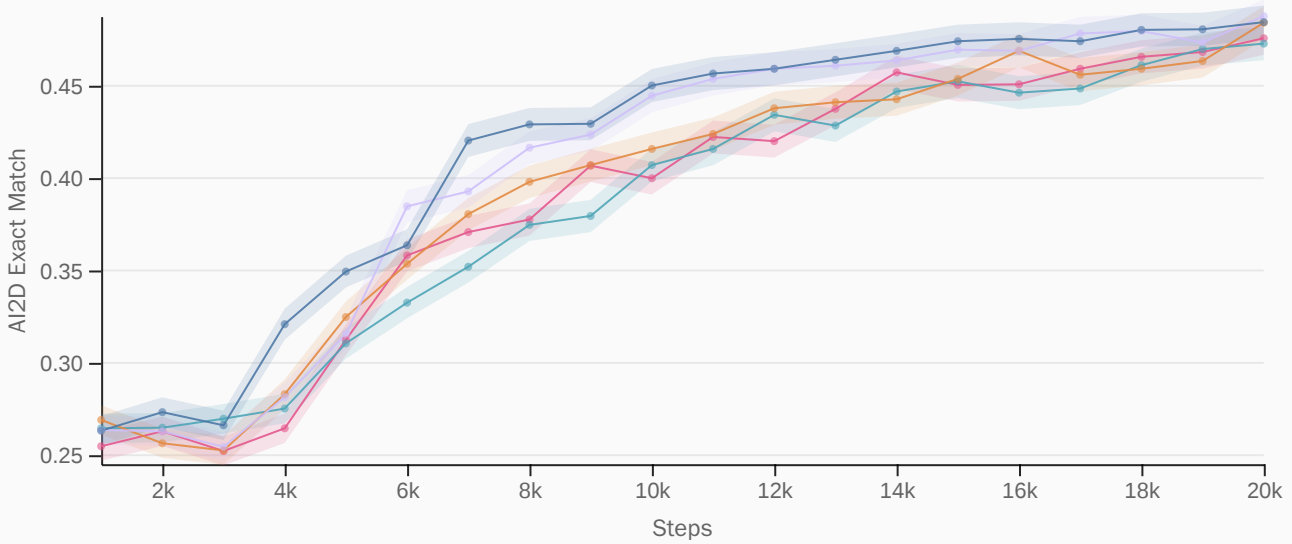
Figure 4: Left: baseline matrix (row-normalized, sequential palette). Right: Δ (Improved − Baseline) in percentage points, using a diverging palette centered at 0 to highlight improvements vs degradations.

d3-line-quad: Comparison across thresholds

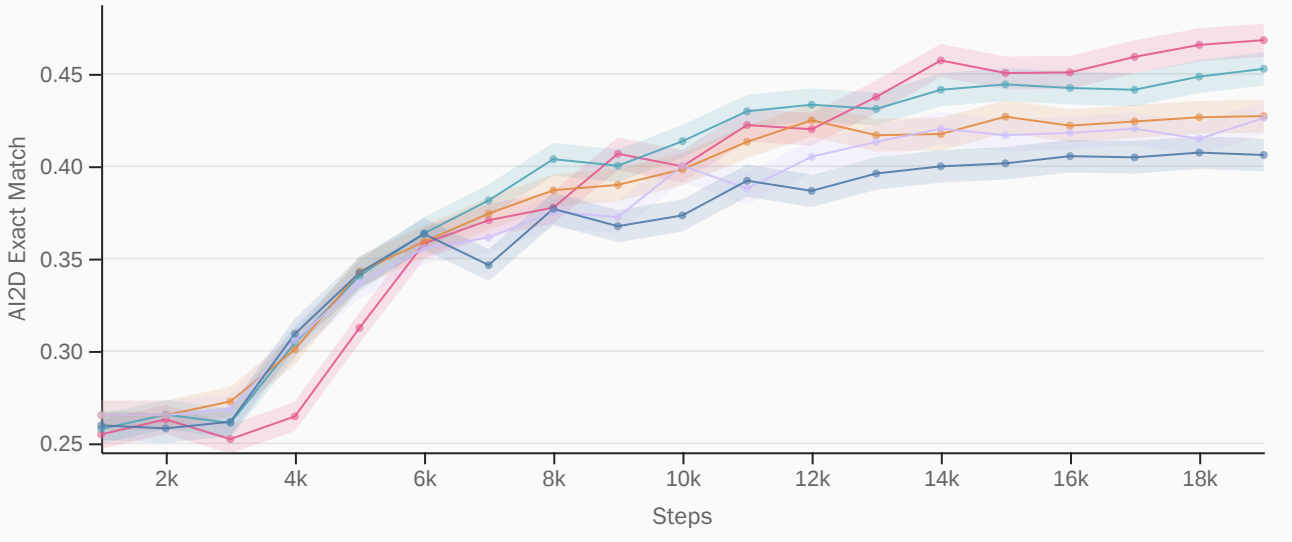
Formatting Filter



Relevance Filter



Visual Dependency Filter



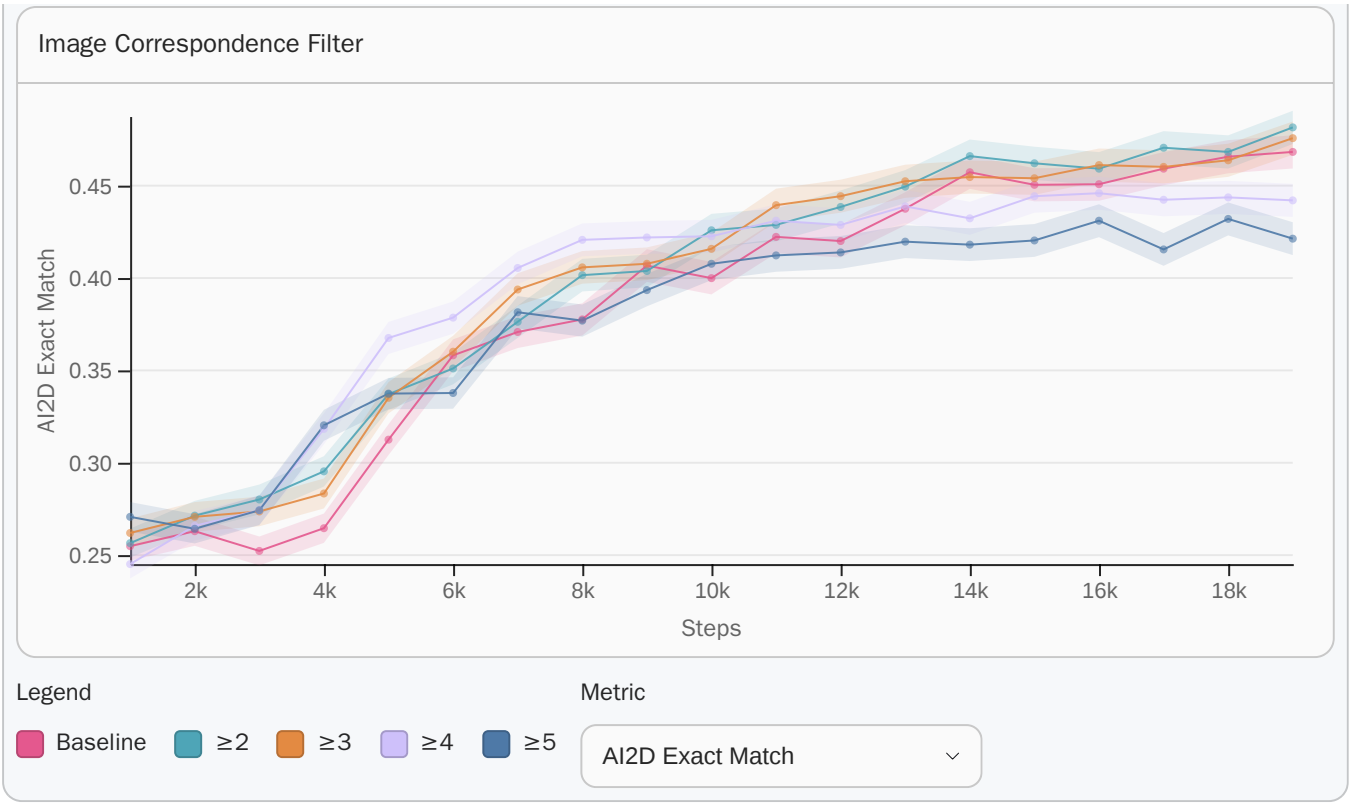


Figure 5: Comparison across thresholds for all four filters individually: Formatting, Relevance, Visual Dependency, and Image-Question Correspondence

Credit: FineVision

d3-bar: Memory usage with recomputation

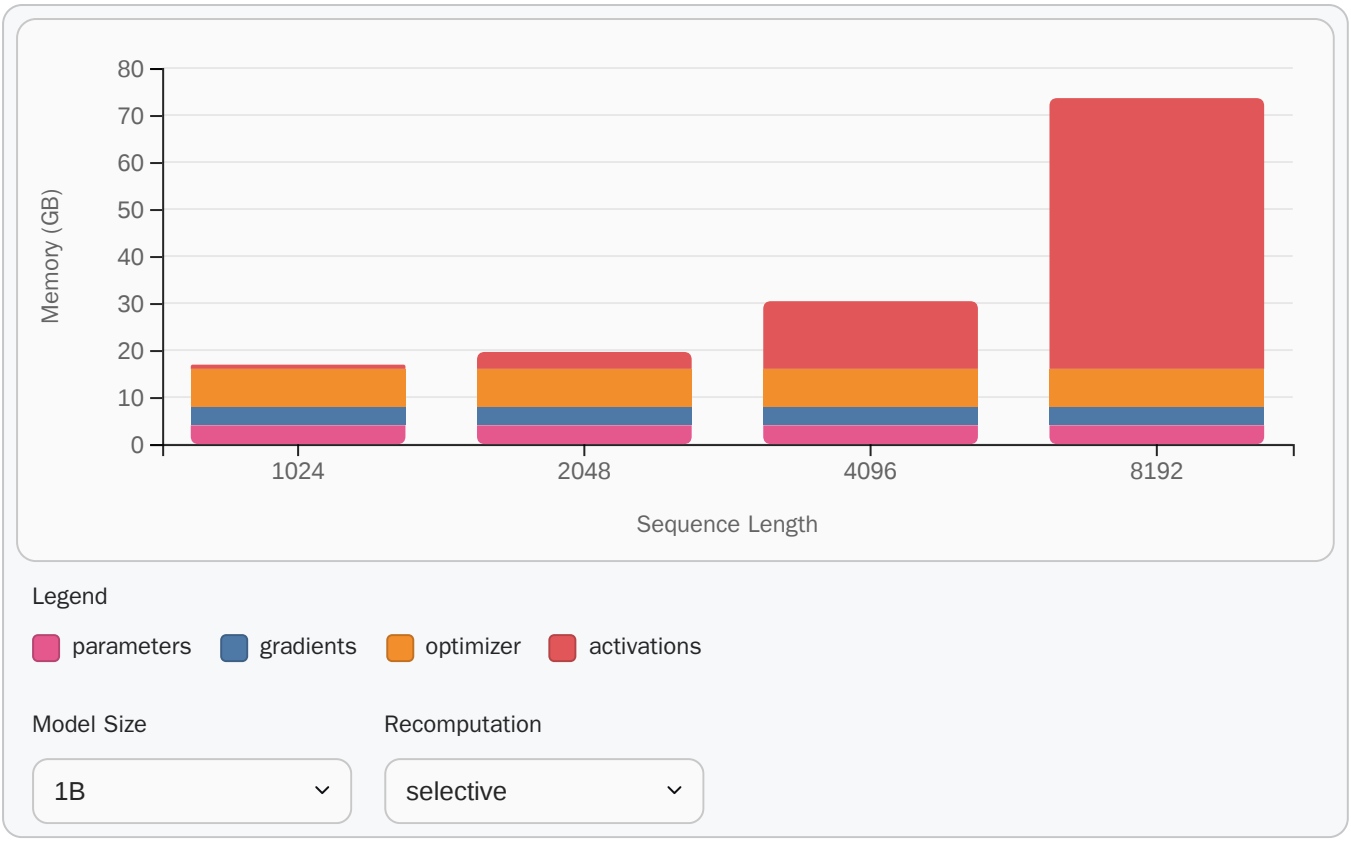


Figure 6: Memory usage with recomputation.

Credits: Ultrascale playbook

d3-pie: Pie charts by category

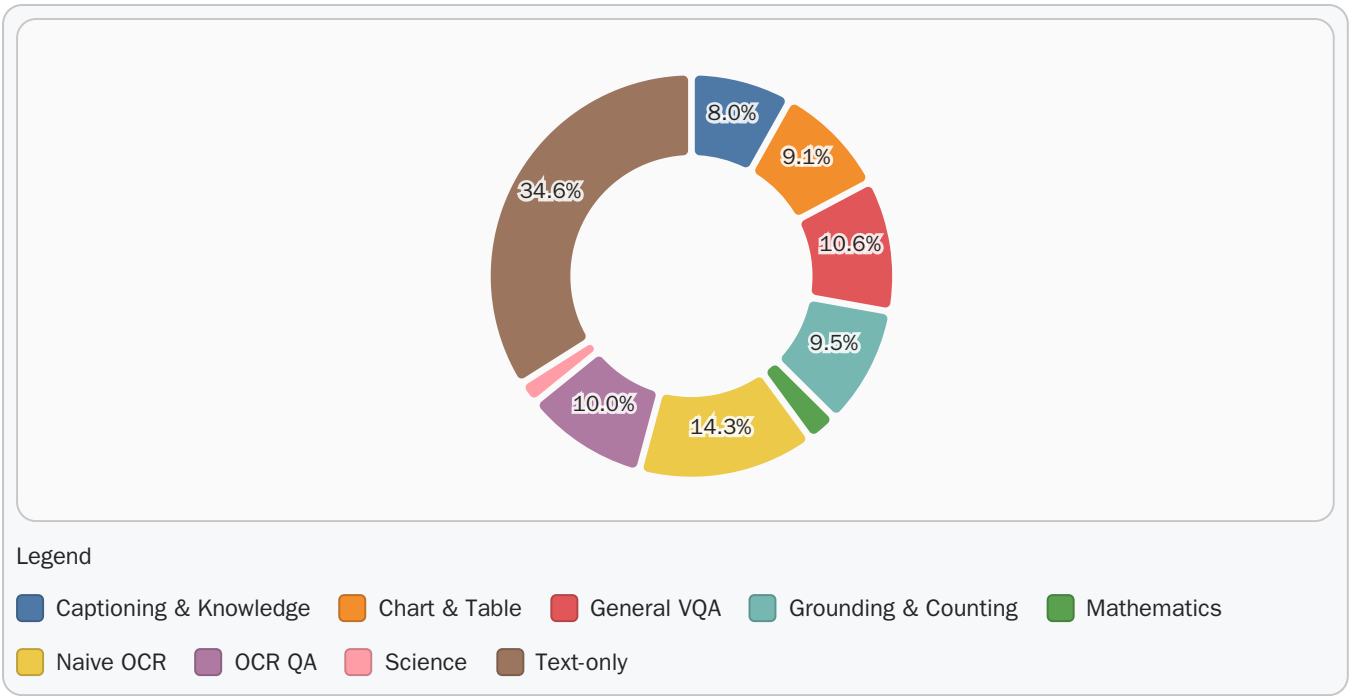
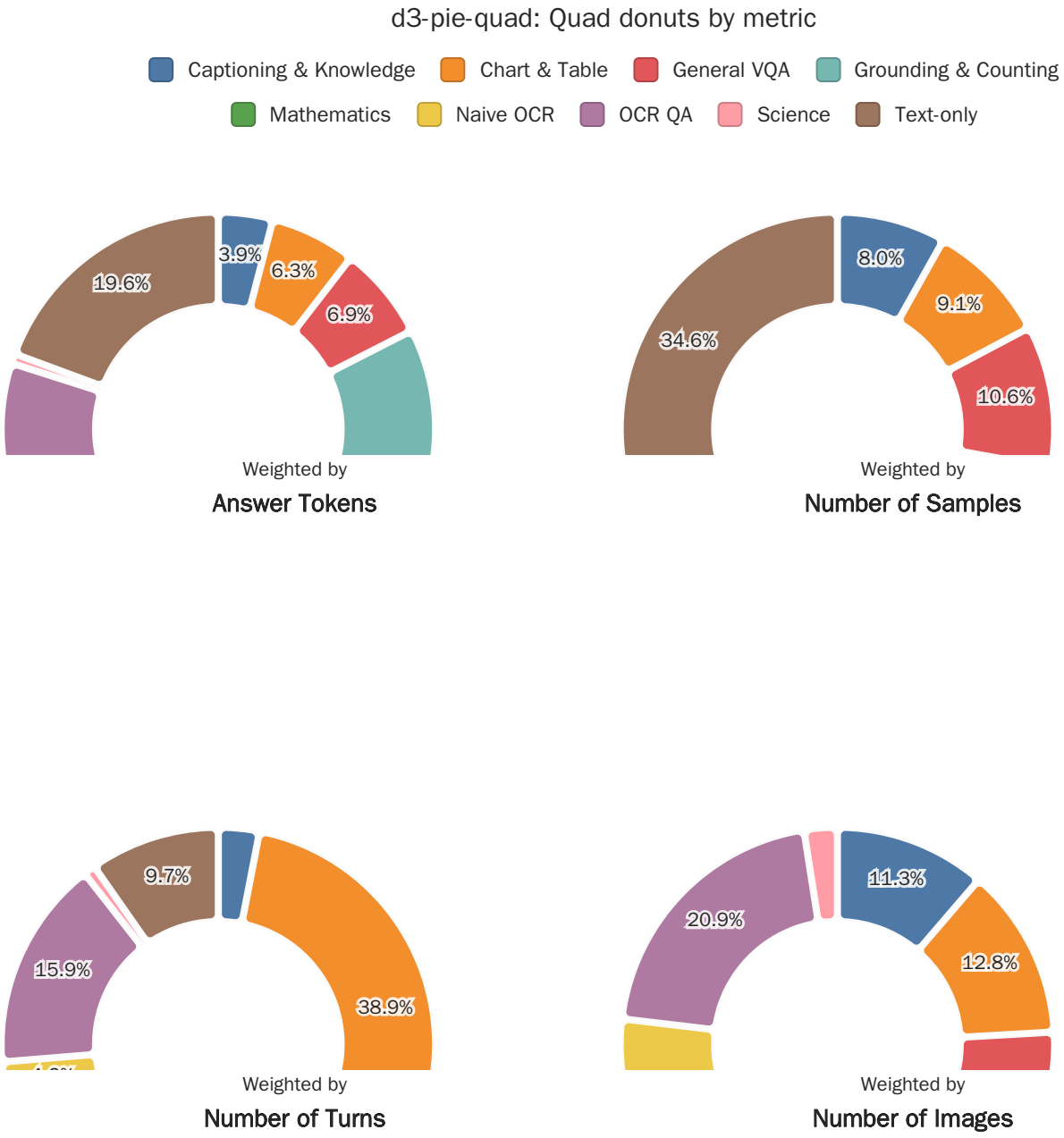


Figure 7: Comparison across thresholds for all four filters individually: Formatting, Relevance, Visual Dependency, and Image-Question Correspondence

Credit: FineVision



Quad view: Answer Tokens, Number of Samples, Number of Turns, Number of Images.

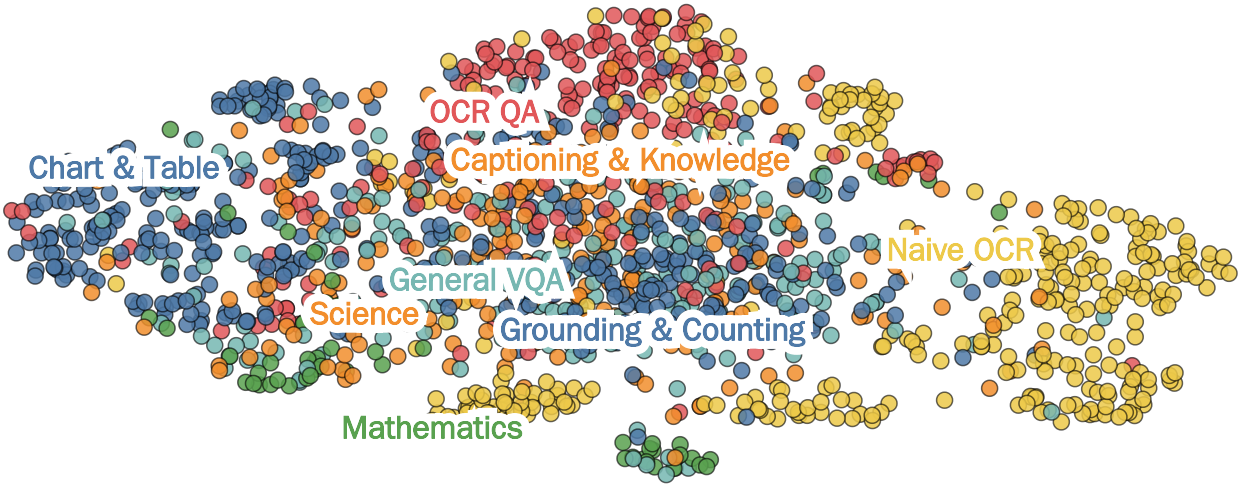
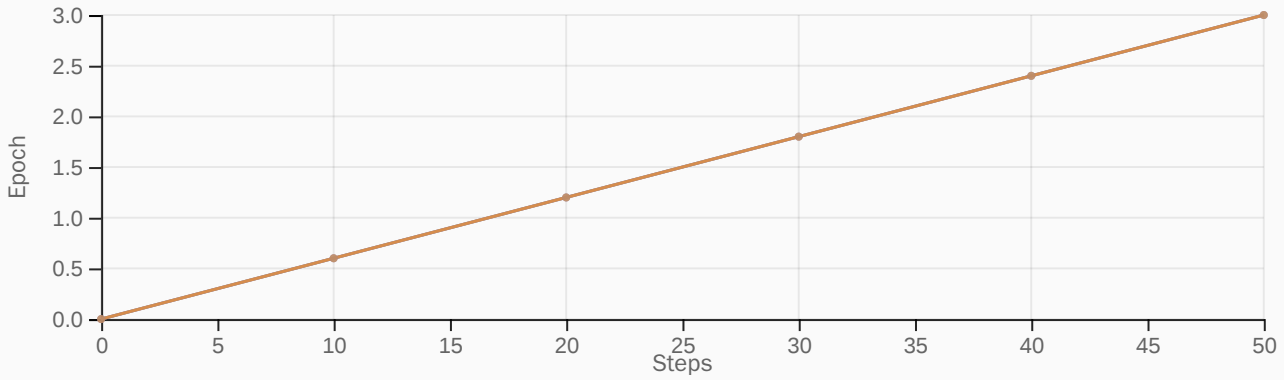


Figure 8: Dataset visualization via UMAP

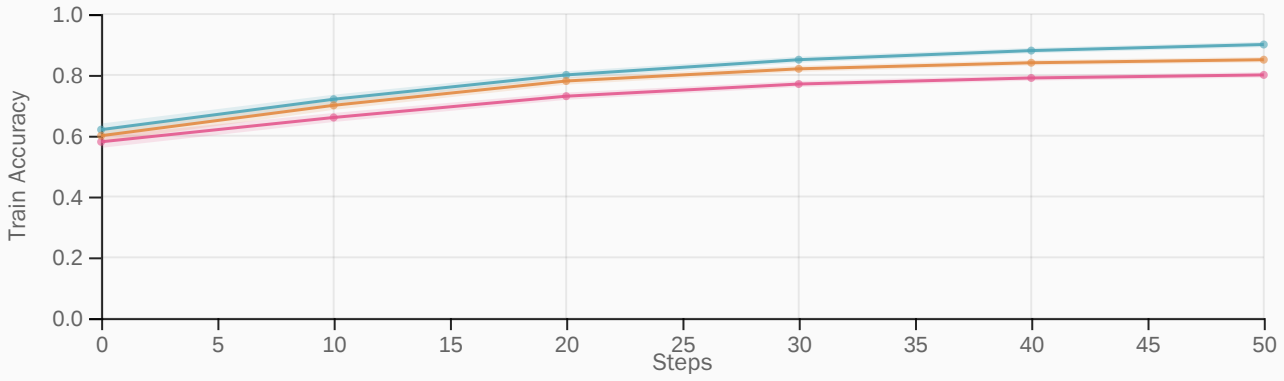
Credit: FineVision

Résultats TrackIO

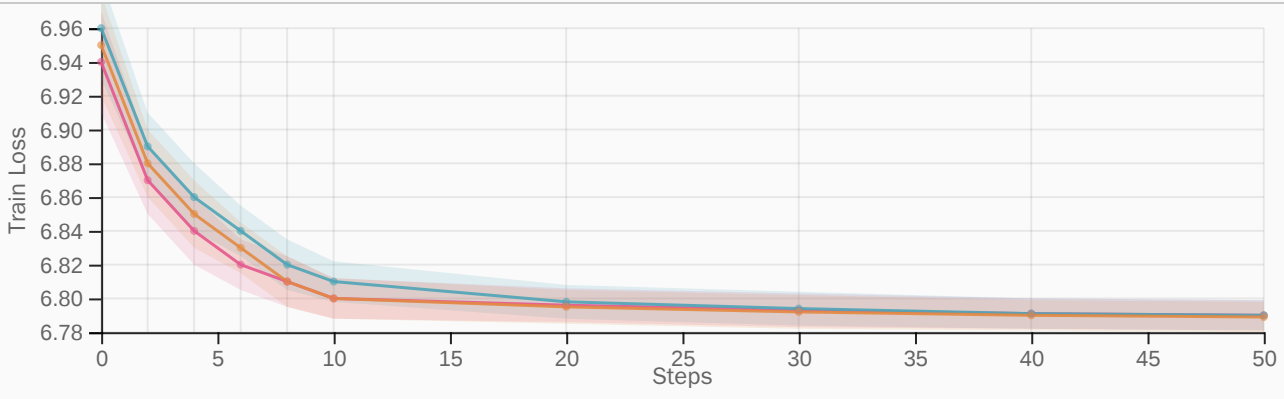
Epoch



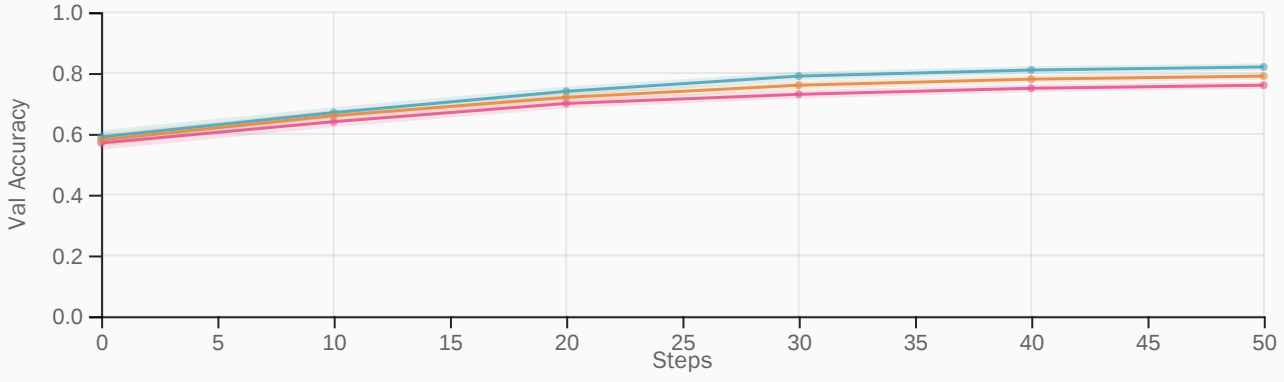
Train Accuracy

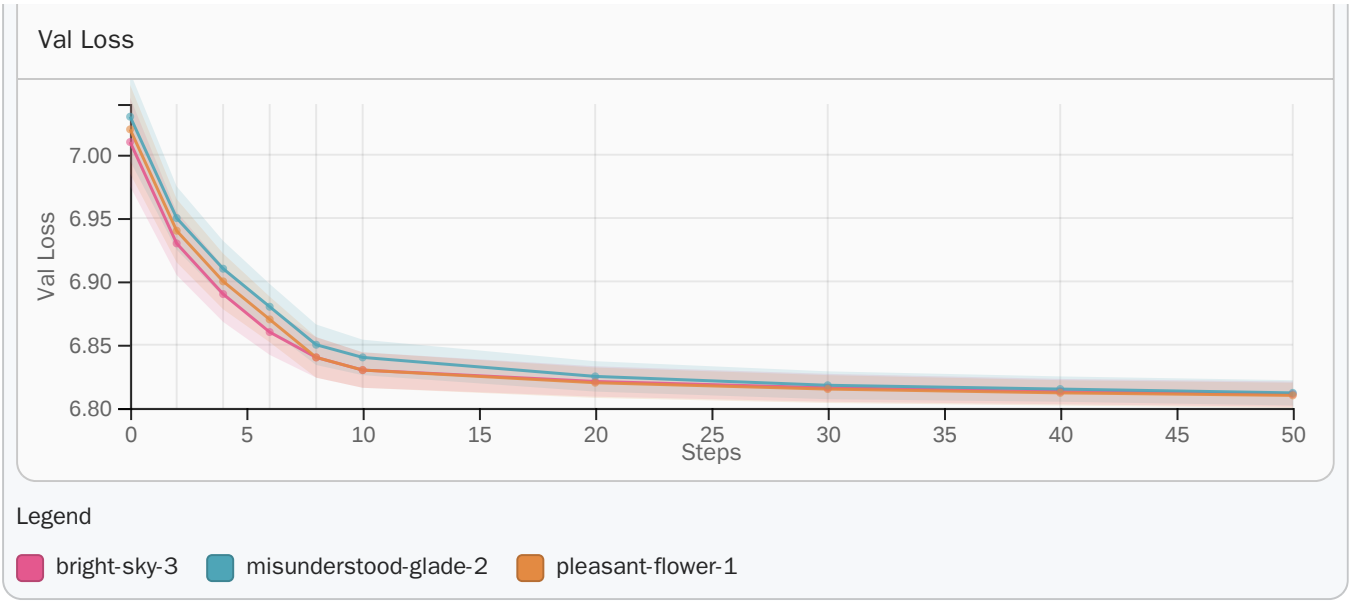


Train Loss



Val Accuracy





Best Practices

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.

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

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,
  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

Reuse

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, Lukasz, & 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).

   ↑