Start here

Short version

- this page explains how the 31-PDF, 1331-page OCR pilot was run and why the final result changed over time
- models tested hands-on in the pilot: GLM-OCR, dots.ocr-1.5, MonkeyOCR, PaddleOCR PP-StructureV3, and FireRed-OCR
- if you only need the workflow routing answer, use the workflow-fit guide: https://instavar.com/blog/ai-production-stack/Which_OCR_Model_Fits_Which_Workflow_in_2026

This page is for the method, the evidence, and the limits.

Trust basis: this was run self-hosted on a single RTX 3090 Ti 24 GB box, the raw outputs were kept, the harness was versioned as it changed, and every page in the corpus was reviewed at contact-sheet scale before the disputed pages were checked again at higher zoom.

The next section gives the gist in under a minute. The later sections move into the method, evidence, and appendices.

If you only have 1 minute

What we tested:

- 31 scanned chemistry PDFs
- 1331 pages
- GLM-OCR, dots.ocr-1.5, MonkeyOCR, PaddleOCR PP-StructureV3, and FireRed-OCR
- one self-hosted workflow on a single 3090 Ti

What changed the result:

- the early runs made GLM-OCR look like the clear default
- that changed only after the FireRed-OCR wrapper was patched to handle near-blank pages and preserve page images
- once that happened, the benchmark stopped being a “one winner” story and became a routing story

The routing rule that survived the audit:

- FireRed-OCR for text-first pages
- GLM-OCR for visual-answer-dependent pages
- dots.ocr-1.5 when the requirement is OCR plus broader visual parsing

What not to overread:

- this is not a universal OCR leaderboard
- this is not a semantic-accuracy proof across all document domains
- this is a scan-heavy, cleanup-oriented benchmark with a visual audit layer

If you have 5 minutes

The pilot started with a practical question:

> On real scanned PDFs, which OCR stack gives the cleanest output for the least downstream cleanup cost, and when should a page be routed to a different model?

That mattered more than asking which model topped a public leaderboard.

What we learned

- the most useful result was not a single winner but a page-routing rule
- FireRed-OCR became the best default on long notes, tables, bullet-heavy pages, and answer-heavy pages
- GLM-OCR stayed safer when the question depends on a local graph, apparatus, reaction scheme, or particle-box option
- dots.ocr-1.5 remained interesting, but more as a broader parser than as the default OCR engine for scan-heavy PDFs

Why the benchmark result changed over time

The early 3-way and 4-way runs were exploratory. The final 5-way result changed because the benchmark harness got better, especially for FireRed-OCR.

Two wrapper fixes mattered:

- a conservative blank-page gate stopped hallucinations on near-empty scans
- page-image preservation stopped figure-heavy pages from being flattened into text-only Markdown

That means the later result is not a mysterious leaderboard reversal. It is a better comparison.

Full decision matrix

Need	Best default from this pilot	Why
Clean Markdown from text-first scanned notes	FireRed-OCR	Lower cleanup burden once blank-page gating and page-image preservation were fixed
Diagram-linked question pages	GLM-OCR	Better inline region preservation for question-local visuals
OCR plus broader visual parsing	dots.ocr-1.5	Better fit when the requirement extends beyond classic document OCR
Mixed PDFs that alternate between notes and figure-heavy worksheet pages	Hybrid routing	Different page types clearly favored different models

Where each model fit

Model	Best fit from this pilot	Main risk if overused
FireRed-OCR	Text-first notes, answer pages, formula-heavy revision pages	Can still lose question-local visuals if a page really depends on them
GLM-OCR	Diagram-dependent question pages	Materially noisier plain-text Markdown on many long note pages
dots.ocr-1.5	OCR plus web, screen, scene, or SVG-style parsing	Not the safest default for scan-heavy school PDFs
PaddleOCR PP-StructureV3	Modular parsing and competitive fallback on messy worksheets	Less clean overall in this Markdown-first comparison
MonkeyOCR	Isolated wins on some pages/documents	Too unstable overall to become the default

What to do with that conclusion

If you are choosing a production OCR workflow, treat this as a routing problem:

- one default for text-first pages
- one safer model for visual-answer-dependent pages
- one optional broader parser if your scope extends beyond document OCR

That is a stronger operational lesson than chasing a single “best” OCR model.

Full methodology

1 What this benchmark was trying to answer

The benchmark question was intentionally narrower than the usual public OCR comparison.

It was not:

- “Which model has the highest OmniDocBench score?”
- “Which model is best in general?”

- “Which model wins on digitally native PDFs?”

It was:

- what happens on scan-heavy notes and worksheets,
- how often the raw output needs cleanup,
- and which failure modes are expensive enough to change the deployment plan.

2 Corpus design

The internal pilot used a narrow but operationally useful slice:

- `31` PDFs
- `1331` pages
- upper-secondary chemistry notes and worksheets
- heavy use of scanned pages, formulas, tables, apparatus diagrams, particle diagrams, answer keys, and mixed question layouts

Three DOCX files in the same source tree were excluded from the OCR bake-off so the comparison stayed PDF-to-PDF and page-for-page.

Why this slice was useful:

- it is difficult enough to expose OCR failure modes quickly
- it mixes text-heavy pages with diagram-dependent question pages
- it is closer to real classroom scan cleanup than to polished benchmark corpora

Why this slice is still limited:

- it is only one domain
- it over-represents scan-heavy school documents
- it should not be treated as a universal proxy for invoices, legal bundles, or clean enterprise PDFs

2.1 A benchmark corpus is not enough without page archetypes

A lot of OCR comparison posts show a few memorable pages without saying which page types are supposed to stress the models. That makes the examples memorable, but it weakens the benchmark.

For future runs, the corpus should always be paired with a smaller archetype suite. The archetype suite is not meant to replace the full corpus. It is meant to make the test explainable.

At minimum, the page archetypes should include:

- text-first notes pages
- diagram-question pages
- table-heavy pages
- formula-heavy pages
- worksheet answer-option pages
- blank or near-blank scans
- noisy or skewed scans if those are part of the deployment brief

Different OCR stacks fail in different ways:

- some lose local diagrams
- some flatten tables
- some break formulas
- some hallucinate on empty pages
- some stay readable on prose but fail once the page becomes option- or figure-dependent

The evaluation should also be reported by slice, not only as one total. That is where slice-level macro-averaging becomes useful.

The cleaner approach is:

- score each archetype slice separately
- compute a macro-average across slices
- still keep the whole-corpus total for operational realism

That prevents a benchmark from looking strong only because it saw many easy text-first pages while quietly failing on diagram-dependent or formula-heavy pages.

2.2 Execution environment

The pilot was run self-hosted on a single `NVIDIA GeForce RTX 3090 Ti` with `24 GB` VRAM.

That was useful for two reasons:

- these were not vendor-hosted black-box OCR calls
- the serving/runtime constraints were close to what a small production team can actually reproduce on one strong workstation

The practical output target was Markdown-first comparison. Where a model exposed more than plain text, those extra artifacts were retained when useful:

- `GLM-OCR`: Markdown plus region-aware outputs
- `FireRed-OCR`: Markdown plus patched page-image preservation in the final run
- `PaddleOCR PP-StructureV3`: full document parsing output normalized into the comparison flow
- `dots.ocr-1.5` and `MonkeyOCR`: raw Markdown-oriented outputs

2.3 Benchmark contract: what has to stay fixed

Models are often shown side by side without keeping the evaluation conditions steady enough for the comparison to mean much.

This pilot did not start with one frozen harness. The final methodology is stricter than the first exploratory runs, and that should be stated directly.

For future runs, the minimum contract should be:

- same page set
- same page order
- same render path for rasterized pages
- same target output format for comparison
- same hardware disclosure
- per-model inference settings documented alongside the output

At a minimum, every run should disclose:

- rendering method and resolution
- OCR prompt or extraction instruction if one exists
- decoding or generation limits if the model is generative
- output target (`Markdown`, `HTML`, `JSON`, or normalized proxy)
- runtime environment
- any post-processing that happens before scoring

Without that contract, the comparison becomes too easy to distort. A model can look stronger or weaker because of truncation limits, low render resolution, missing blank-page handling, or a quiet change in post-processing.

That is exactly why the patched `5`-way comparison is more trustworthy than the earlier exploratory passes.

2.4 Benchmark changelog and run versioning

OCR write-ups often present every comparison as though it came from one perfectly frozen harness. That was not true in this pilot, and it should not be hidden.

For each refresh, the workflow should log:

- corpus version
- archetype-suite version
- model versions
- inference settings
- post-processing steps
- scoring logic version
- any wrapper fixes that change the outcome materially

That is not bureaucracy. It is the only clean way to explain why a later run changed the ranking.

In this pilot, the practical example was straightforward:

- the early `3`-way and `4`-way runs were exploratory
- the later `5`-way run changed because the `FireRed-OCR` wrapper was patched
- the ranking shift was therefore a pipeline change, not a mysterious benchmark reversal

3 Model set and run sequence

The exploration did not start with five models at once. It expanded over time as the failure modes became clearer.

3.1 Public benchmark families we track

Before running any internal bake-off, it helps to separate public benchmark families by what they are actually measuring.

The three families worth tracking most closely are:

- `OmniDocBench` for broad document parsing quality across complex page structures
- `OlmOCR-Bench` for narrower OCR-style checks and unit-test-like failure cases
- `CC-OCR` for multilingual and broader cross-category OCR coverage

Those benchmark families are useful, but they should not be flattened into one fake universal leaderboard.

In practice:

- `OmniDocBench` is useful for broad document understanding directionally
- `OlmOCR-Bench` is useful for precise failure-style comparisons
- `CC-OCR` is useful when multilingual or cross-domain OCR behavior matters

What they do **not** do is replace a fixed in-house bake-off on the page types you actually deploy against.

3.2 Run history

The pilot expanded in three stages.

Raw 3-way comparison

Models:

- `GLM-OCR`
- `dots.ocr-1.5`
- `MonkeyOCR`

Headline result:

- wins: `GLM=24`, `dots=2`, `Monkey=5`

Model totals inside that run:

Model	Artifact score total	Notes
GLM-OCR	`2117`	Clear early default on this corpus
dots.ocr-1.5	`5750`	Hurt by hallucinations and watermark-like residue in that raw pass
MonkeyOCR	`4696`	Some strong page/document wins, but unstable overall

What that clarified:

- `GLM-OCR` was the safest early default on this scan-heavy corpus
- `dots.ocr-1.5` was interesting, but not the safest baseline for pure scanned-document OCR
- `MonkeyOCR` produced some strong document wins, but it was not stable enough to become the default

Raw 4-way comparison

Models:

- `GLM-OCR`
- `dots.ocr-1.5`
- `MonkeyOCR`
- `PaddleOCR PP-StructureV3`

Headline result:

- wins: `GLM=14`, `dots=4`, `Monkey=3`, `Paddle=10`

Model totals inside that run:

Model	Artifact score total	Notes
GLM-OCR	`4695`	Strongest overall baseline in that `4`-way pass
dots.ocr-1.5	`6098`	Still a weak default for this scan-heavy OCR slice
MonkeyOCR	`6755`	Too noisy overall despite some wins
PaddleOCR PP-StructureV3	`5827`	Much more competitive than a paper-only read would suggest

What that clarified:

- `PaddleOCR` was more competitive than a public-paper-only reading would suggest, especially on some messy worksheet-style documents
- `GLM-OCR` still held the strongest overall baseline position
- `dots.ocr-1.5` remained more compelling as a broader visual parsing candidate than as the default scanned-document OCR engine

Patched 5-way comparison

Models:

- `GLM-OCR`
- `dots.ocr-1.5`
- `MonkeyOCR`
- `PaddleOCR PP-StructureV3`
- `FireRed-OCR`

Before this run, the `FireRed-OCR` wrapper was patched in two specific ways:

- a conservative blank-page gate to stop hallucinations on near-empty pages
- page-image preservation so figure-heavy pages were not silently flattened into text-only Markdown

Headline result:

- wins: `FireRed=24`, `GLM=2`, `dots=2`, `Monkey=1`, `Paddle=2`

Final `5`-way artifact totals:

Model	Artifact score total	Notes
FireRed-OCR	`2215`	Lowest total in the patched run; `426` page-image refs preserved
GLM-OCR	`4655`	Strong on diagram-linked pages, but materially noisier on many text-first pages

PaddleOCR PP-StructureV3	`5787`	Competitive on some messy pages, but less clean overall in this run
dots.ocr-1.5	`6053`	Better read as a broader visual parser than a pure OCR default
MonkeyOCR	`6715`	Some document wins, but weakest total in the final run

Important caveat:

- the `3`-way, `4`-way, and patched `5`-way runs are all useful
- the patched `5`-way result should be read as the final routing-oriented comparison because it reflects the fixed `FireRed-OCR` evaluation path

Another caveat matters for trust: do not read the absolute totals from the early raw runs and the patched `5`-way run as one single frozen leaderboard. The harness evolved during the pilot. The useful signals are:

- within-run rankings
- named failure modes
- whether the ranking changed for a defensible pipeline reason

4 Comparison dimensions we score before choosing a model

Public OCR roundups often collapse everything into one question: “which model is best?”

That is not how OCR decisions should be made.

Before promoting any stack, the comparison should at least score or describe these dimensions:

Dimension	Why it matters	What to check in practice
Text fidelity	Raw text still has to be right before anything else matters	Missing lines, substitutions, duplicated spans, truncation
Structural integrity	Markdown, HTML, or JSON breakage directly affects cleanup cost	Broken lists, malformed tables, invalid nesting, broken formula blocks
Locality / grounding	Some pages depend on small local visuals, not just page-level text	Inline diagram preservation, region references, question-local figures
Output format fit	The right output depends on downstream use, not just OCR quality	Markdown for editing, HTML for layout retention, JSON for extraction
Table behavior	Tables often fail differently from prose	Row and column closure, merged cells, header retention, readable export
Formula behavior	Formula corruption can make science and technical documents unusable	LaTeX validity, symbol substitutions, inline versus block math stability
Blank-page behavior	Hallucinations on empty scans can distort benchmark results	Empty-page skip, near-blank gating, false positives
Runtime / serving path	A good model that is awkward to serve may still be the wrong choice	Single-GPU fit, batch throughput, wrapper complexity, memory use
Domain fit	Benchmarks underrepresent many real document types	Notes, worksheets, forms, receipts, bank statements, multilingual scans

This is one reason the final routing rule ended up more useful than a single winner. Different models were stronger on different dimensions, and the deployment choice became a routing problem rather than a one-model problem.

5 Scoring and visual audit

The comparison needed something faster than full line-by-line human adjudication across `1331` pages, but more useful than word count alone.

5.1 Artifact score

The main heuristic was an `artifact score`.

Lower is better.

The score penalized:

- duplicate non-empty lines
- spaced-letter OCR artifacts
- fused-token OCR artifacts
- hallucination lines
- watermark-like residue

The scoring logic used this shape:

```
artifact_score =
  duplicate_nonempty_lines
  + 2 * spaced_letter_artifacts
  + 2 * fused_token_artifacts
  + 5 * hallucination_lines
  + 3 * watermark_hits
```

Why this was useful:

- it surfaces cleanup cost
- it catches obvious OCR formatting damage quickly
- it gives a stable first pass for comparing raw Markdown output at scale

Why it is not enough on its own:

- it is not a semantic-accuracy percentage
- it does not fully capture lost diagrams
- it can over-penalize or under-penalize edge cases if used without manual review

So the benchmark did not stop at the heuristic.

5.2 Manual visual audit layer

The heuristic pass was followed by visual review.

Audit method:

- full `31`-PDF corpus reviewed at contact-sheet scale across all `1331` pages
- deeper zoomed page-level passes on the two remaining `GLM-OCR` wins
- deeper zoomed passes on three ambiguous mixed documents that could not be classified confidently from the contact sheets alone

Every page in the corpus was reviewed visually at contact-sheet scale, and the important disagreements were checked again at higher zoom before any routing conclusion was kept.

The visual audit should publish a compact review packet for the pages that changed the routing policy:

- one source-page render
- one raw output excerpt from each model being discussed
- one sentence on the decisive failure mode
- one routing conclusion tied to that page type

6 Failure modes that actually changed the routing

The benchmark became useful only once the failure modes were named clearly.

6.1 Blank-page hallucination

This mattered most for `FireRed-OCR` early on.

Near-empty scanned pages triggered hallucinated content until the workflow added a conservative blank-page gate. Without that fix, FireRed looked worse than it really was.

Lesson:

- evaluation pipelines can distort model quality if they do not treat blank or near-blank scans explicitly

6.2 Inline diagram loss

This is where `GLM-OCR` stayed valuable.

When a page contains:

- a reaction scheme
- an apparatus sketch
- particle-box answer options
- a graph that the question directly references

the OCR problem is not just “read the text.” It is “keep the local visual attached to the question.”

`GLM-OCR` remained safer on those pages because its inline region preservation was more useful than cleaner plain-text output.

6.3 Text-cleanliness and cleanup cost

Once blank-page handling was fixed, `FireRed-OCR` often produced cleaner Markdown on:

- explanatory notes
- answer pages
- formula-heavy sections
- bullet-heavy revision pages
- long text-first classroom notes

Cleanup time is often the hidden cost center in OCR projects.

6.4 Broader parser vs better OCR default

`dots.ocr-1.5` stayed interesting, but for a different reason.

It is more valuable when the requirement includes:

- web or UI parsing
- scene text
- SVG-style outputs
- broader visual parsing beyond classic document OCR

That is a genuine product difference. It is not the same decision as picking the safest OCR default for scanned PDFs.

6.5 The early harness and the final harness were not identical

This is worth stating plainly because credibility depends on it.

The raw `3`-way and `4`-way runs were exploratory passes. The patched `5`-way run came later, after the pilot fixed specific evaluation weaknesses, especially around `FireRed-OCR`.

That means:

- the early runs were still useful
- the final run is the most trustworthy routing-oriented comparison
- the real story is not “one model flipped the leaderboard overnight”

The real story is that the benchmark got better once the pipeline handled blank pages and figure preservation more honestly.

7 Concrete documents that changed the routing policy

The benchmark would have been much less useful if it only ended with aggregate win counts.

These documents are the ones that actually changed the policy.

7.1 GLM-only case

`Topic 1B Identification of Ions and Gases` stayed a `GLM-OCR` document.

Why:

- almost every page depended on small inline reaction schemes or figure-linked question content
- page-level image links were not enough because the diagram had to stay tied to a specific question

This was the clearest case where cleaner text alone was not the right success metric.

7.2 Hybrid cases

These documents were mixed enough that whole-document routing lost quality:

- `Elements, Compounds and mixtures notes`
- `Kinetic Particle Notes`
- `Elements, Compounds and Mixtures Worksheet`
- `Speed of Reaction Worksheet`

The shared pattern was consistent:

- question pages with graph, apparatus, or particle-box answer options leaned `GLM-OCR`
- worked answers, explanation pages, and text-heavy notes leaned `FireRed-OCR`

7.3 FireRed-heavy wins

The documents that made the `FireRed-OCR` case sharper were the text-heavy note packs and answer-heavy pages, including examples such as:

- `Acids and Bases Notes`
- `Ionic Bonding Notes`
- `Energy Changes`
- `Electrolysis`
- `The Atmosphere and Environment`

Those files made it clear that once blank-page hallucinations were controlled, FireRed often reduced the cleanup burden on long note pages materially.

8 The routing rule that survived the audit

The most useful result from the benchmark was a routing rule that generalizes better than “best model.”

Use:

- `FireRed-OCR` for text-first pages
- `GLM-OCR` for visual-answer-dependent pages
- `dots.ocr-1.5` when the real requirement extends beyond document OCR into broader visual parsing

In practice, the page types that leaned `GLM-OCR` were:

- “the diagram below” questions
- apparatus questions
- graph-selection questions
- particle-box answer choices
- local reaction schemes
- MCQ pages where the options themselves are images

The page types that leaned `FireRed-OCR` were:

- explanatory notes
- bullet lists
- worked answers
- answer keys
- tables

- formula-heavy pages
- long prose-heavy revision notes

The mixed-document outcome matters too:

- some PDFs should not be routed whole-doc to one model
- a single file can alternate between text-heavy notes and figure-dependent worksheet pages

What this benchmark does not prove

This benchmark does **not** prove that `FireRed-OCR` is the best OCR model in general.

It also does **not** prove that `GLM-OCR` is weak, or that `dots.ocr-1.5` is a poor release.

It proves something narrower and more operationally useful:

- on one hard scan-heavy corpus
- with this scoring method
- and with manual visual audit layered on top
- the routing rule mattered more than the public leaderboard story

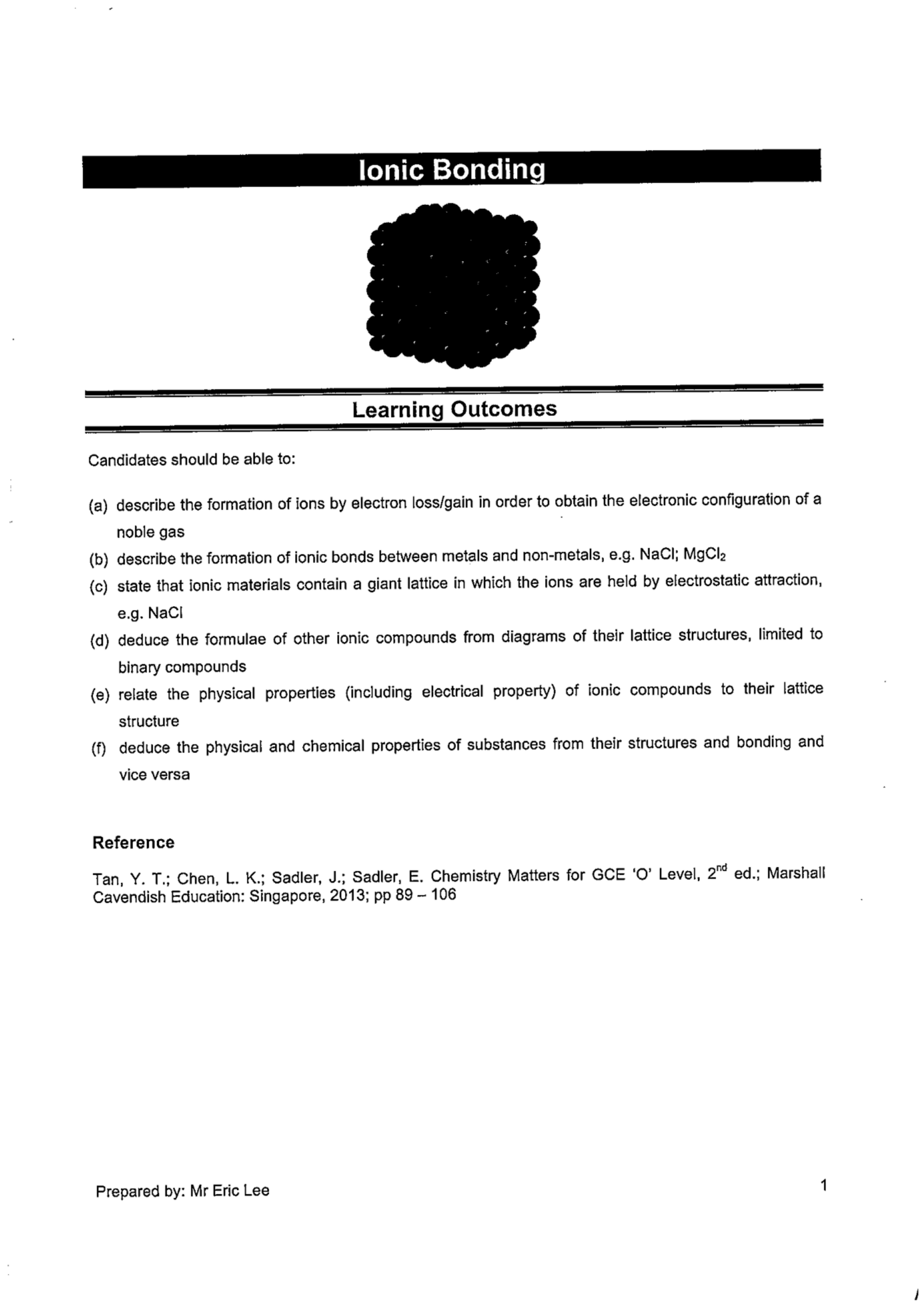
That is a better deployment lesson than chasing one winner.

Appendix A: Concrete page evidence from the pilot

Aggregate totals were useful, but the routing rule only became credible once the decisive page types were visible.

The examples below are taken from the actual scan-heavy pilot. They are not polished post-processed outputs. The point is to show what the models did before editorial cleanup.

Example 1: FireRed on a text-first notes page



Ionic Bonding

Learning Outcomes

Candidates should be able to:

(a) describe the formation of ions by electron loss/gain in order to obtain the electronic configuration of a noble gas

(b) describe the formation of ionic bonds between metals and non-metals, e.g. $NaCl$; $MgCl_2$

(c) state that ionic materials contain a giant lattice in which the ions are held by electrostatic attraction, e.g. $NaCl$

(d) deduce the formulae of other ionic compounds from diagrams of their lattice structures, limited to binary compounds

(e) relate the physical properties (including electrical property) of ionic compounds to their lattice structure

(f) deduce the physical and chemical properties of substances from their structures and bonding and vice versa

Reference

Tan, Y. T.; Chen, L. K.; Sadler, J.; Sadler, E. Chemistry Matters for GCE 'O' Level, 2nd ed.; Marshall Cavendish Education: Singapore, 2013; pp 89 – 106

This page is mostly linear text with a simple formula mention. That is where `FireRed-OCR` tended to reduce cleanup burden.

FireRed excerpt:

(b) describe the formation of ionic bonds between metals and non-metals, e.g. $NaCl$; $MgCl_2$

GLM excerpt:

```
(b) describe the formation of ionic bonds between metals and non–metals, e.g. NaCl; $ \mathrm{M g C l\_{2}} $
```

That is a small example, but it matters. On long note pages, repeated chemistry-token damage adds up quickly.

Example 2: GLM on an inline reaction-scheme question

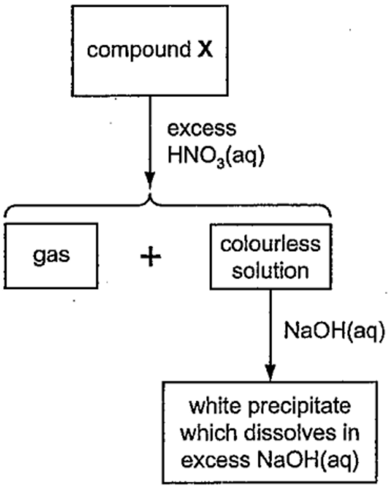
Topic 1B Identification of Ions and Gases

PAPER 1

MULTIPLE-CHOICE QUESTIONS

For each question, there are four possible answers. Choose the one you consider correct and record your choice (A, B, C or D) in the brackets provided.

1. The diagram shows a reaction scheme for compound **X**.



What is compound **X**? (N2008/P1/Q2)

A aluminium sulphate **B** calcium carbonate

C copper(II) carbonate **D** zinc carbonate ( )

2. Excess aqueous sodium hydroxide is added to salt X and the solution is heated. A gas is given off which turns red litmus blue.
When this reaction is complete, aluminium foil is added to the solution.
A gas is again given off which also turns red litmus blue.
What is salt X? (N2009/P1/Q1)
A ammonium nitrate
B ammonium sulfate
C zinc nitrate
D zinc sulfate ( )

3. An aqueous solution of compound X reacts with aqueous sodium hydroxide to form a green precipitate and then aluminium powder is added. The mixture is heated and a gas that turns damp red litmus paper blue is given off.
What is X? (N2010/P1/Q2)
A ammonium nitrate **B** copper(II) chloride
C iron(III) chloride **D** iron(II) nitrate ( )

This is the page type where GLM-OCR stayed safer. The question is not only about text. It depends on the local reaction scheme.

GLM excerpt:

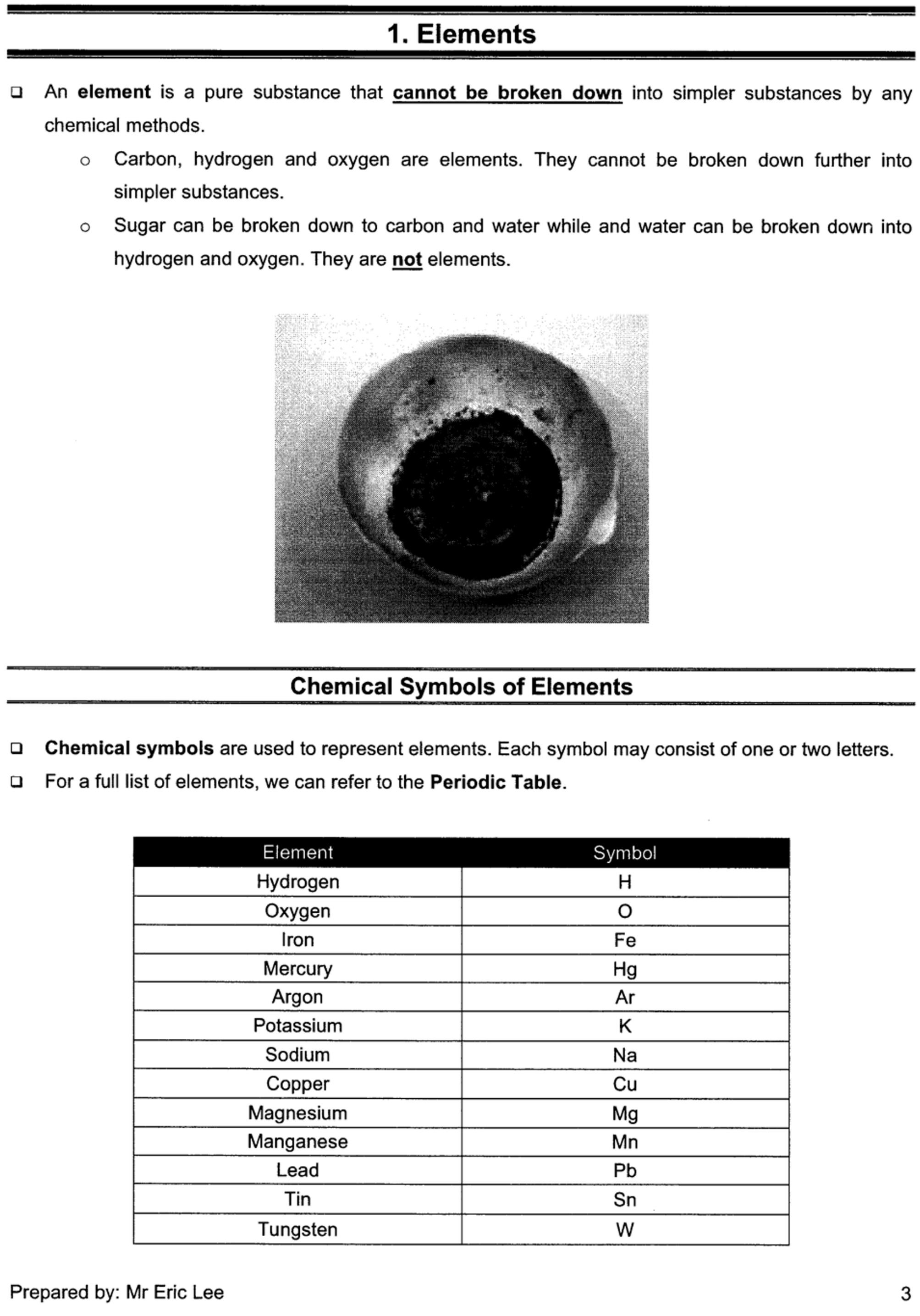
```
1. The diagram shows a reaction scheme for compound X.
   ![Image 0-0](imgs/cropped_page0_idx0.jpg)
   What is compound X? (N2008/P1/Q2)
```

FireRed excerpt:

```
1. The diagram shows a reaction scheme for compound X.
   What is compound X?
   A aluminium sulphate
   B calcium carbonate
   C copper(II) carbonate
   D zinc carbonate
```

The FireRed text is readable, but the question-local diagram is no longer preserved inline. So this document stayed `GLM-OCR`.

Example 3: Why mixed documents need hybrid routing, part 1



1. Elements

- An **element** is a pure substance that **cannot be broken down** into simpler substances by any chemical methods.
  - Carbon, hydrogen and oxygen are elements. They cannot be broken down further into simpler substances.
  - Sugar can be broken down to carbon and water while and water can be broken down into hydrogen and oxygen. They are **not** elements.

Chemical Symbols of Elements

- **Chemical symbols** are used to represent elements. Each symbol may consist of one or two letters.
- For a full list of elements, we can refer to the **Periodic Table**.

Element	Symbol
Hydrogen	H
Oxygen	O
Iron	Fe
Mercury	Hg
Argon	Ar
Potassium	K
Sodium	Na
Copper	Cu
Magnesium	Mg
Manganese	Mn
Lead	Pb
Tin	Sn
Tungsten	W

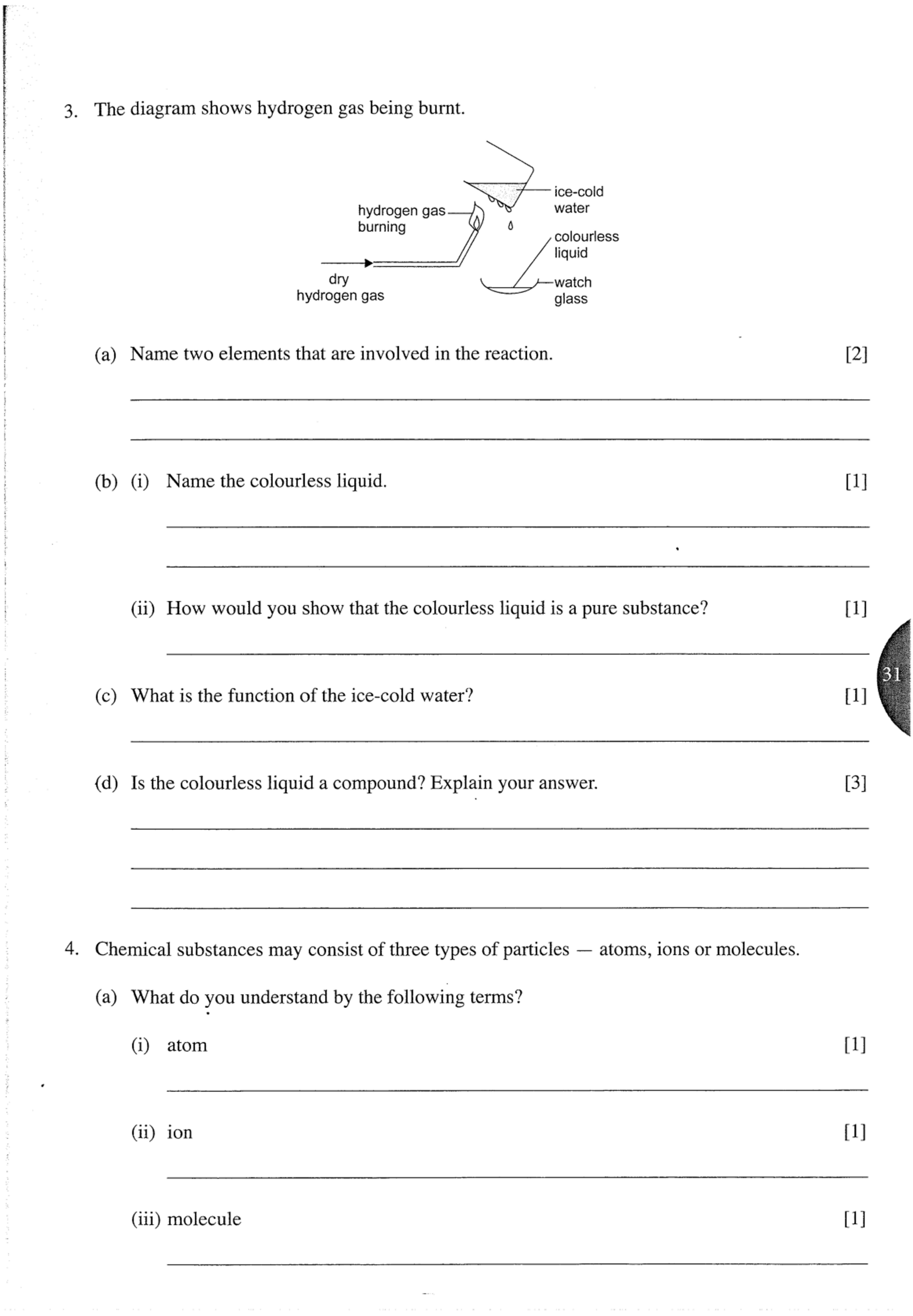
This page comes from one of the documents that ended up in the hybrid bucket rather than the `GLM-only` bucket.

On text-and-table pages like this one, FireRed was already good enough to be the better operational choice:

```
### 1. Elements

□ An element is a pure substance that cannot be broken down into simpler substances by any chemical methods.

## Chemical Symbols of Elements

<table><tr><td>Element</td><td>Symbol</td></tr><tr><td>Hydrogen</td><td>H</td></tr><tr><td>Oxygen</td><td>O</td></tr>...
```

That is also why the document did not stay on `GLM-OCR` wholesale. Some pages were already cleaner and easier to work with in FireRed.

Example 4: Why mixed documents need hybrid routing, part 2



3. The diagram shows hydrogen gas being burnt.

hydrogen gas burning
ice-cold water
colourless liquid
dry hydrogen gas
watch glass

(a) Name two elements that are involved in the reaction. [2]

(b) (i) Name the colourless liquid. [1]

(ii) How would you show that the colourless liquid is a pure substance? [1]

(c) What is the function of the ice-cold water? [1]

(d) Is the colourless liquid a compound? Explain your answer. [3]

4. Chemical substances may consist of three types of particles — atoms, ions or molecules.

(a) What do you understand by the following terms?

(i) atom [1]

(ii) ion [1]

(iii) molecule [1]

The same document later switches into a diagram-linked worksheet page. This is where GLM-OCR was safer.

GLM excerpt:

```
3. The diagram shows hydrogen gas being burnt.
   ![Image 14-22](imgs/cropped_page14_idx22.jpg)
   ( a ) Name two elements that are involved in the reaction. [2]
   ...
4. Chemical substances may consist of three types of particles – atoms, ions or molecules.
   ...
   ![Image 14-23](imgs/cropped_page14_idx23.jpg)
```

FireRed excerpt:

```
3. The diagram shows hydrogen gas being burnt.
   (a) Name two elements that are involved in the reaction. [2]
   (b) (i) Name the colourless liquid. [1]
   ...
4. Chemical substances may consist of three types of particles – atoms, ions or molecules.
   (a) What do you understand by the following terms?
   (i) atom
   (ii) ion
```

This is the same PDF as Example 3. One page type leans `FireRed-OCR`; the other leans `GLM-OCR`. Hybrid routing was not an edge case here. It was a real deployment requirement.

Example 5: Why benchmark versioning matters

This page is the clearest example of why run versioning belongs in the methodology itself.

Earlier FireRed run:

```
<!-- page 58 -->

The 7 Habits of Highly Effective People
```

Patched FireRed run:

```
<!-- page 58 -->
<!-- page 58 blank-page-skipped -->
```

The model did not become magically different overnight. The wrapper changed. That is exactly why benchmark changelogs and run-versioning have to be explicit.

Appendix B: Full workflow to reuse the method

If an OCR team wants a practical first benchmark instead of a benchmark-shaped press release, a sensible process is:

1. Build a corpus that mixes:
   - text-first pages
   - diagram-dependent pages
   - tables
   - formulas
   - answer keys
   - blank or near-blank scans
2. Define a smaller archetype suite so the benchmark has named page types, not only a blob of documents.
3. Lock the benchmark contract:
   - same pages
   - same render path
   - same output target
   - per-model settings disclosed
4. Run the same pages through every candidate stack.
5. Score cleanup-oriented artifacts, not just word count.
6. Publish raw page outputs for the pages that actually changed the decision.
7. Add visual review before promoting any model.
8. Treat routing as part of the product, not as a post-hoc patch.

That is the main lesson from this pilot.

Appendix C: Related OCR reading

- broader market map: https://instavar.com/blog/ai-production-stack/OCR_SOTA_Feb_2026_Open_Document_AI_Leaderboard
- workflow-fit guide: https://instavar.com/blog/ai-production-stack/Which_OCR_Model_Fits_Which_Workflow_in_2026