

# PaperQuay: A Desktop-First Environment for Structured Academic Reading

PaperQuay Onboarding Document

## Abstract

PaperQuay is a lightweight desktop paper reader designed for researchers, students, and knowledge workers who spend significant time reading academic PDFs. Instead of treating a paper as a flat document, PaperQuay treats it as a structured reading workspace. A paper can be connected to a local library, opened as a PDF, parsed into document blocks, translated at the block level, summarized by AI, and revisited later with its reading state preserved.

The main idea behind PaperQuay is simple: academic reading is not only about displaying pages. A good reading environment should help the reader move between the original PDF layout, extracted semantic content, translation, notes, and AI-assisted understanding. PaperQuay therefore combines a traditional PDF viewer with structured block extraction, linked navigation, local workspace management, and optional AI features.

## 1 Introduction

Academic papers are usually distributed as PDF files. The PDF format preserves visual layout very well, but it is not always ideal for deep reading. When a reader wants to understand a complex section, compare a figure with its explanation, translate a paragraph, or ask questions about the paper, a plain PDF viewer provides limited support. The reader often has to switch between several tools: a PDF viewer, a translation service, a note-taking application, a reference manager, and a chat-based AI assistant.

PaperQuay is designed to reduce this fragmentation. It provides a single desktop workspace where the original PDF remains visible while additional structured information is placed around it. The reader can start from a local Zotero library, open a paper, run MinerU parsing, inspect extracted blocks, translate content, generate summaries, and ask questions about the current paper. The application is desktop-first, which means local file paths, local cache, and local paper workspaces are treated as core parts of the design rather than as afterthoughts.

The goal is not to replace Zotero, nor to replace a professional annotation system. Instead, PaperQuay focuses on the reading moment: the period when a user has selected a paper and wants to understand it efficiently, accurately, and repeatedly.

## 2 Design Philosophy

PaperQuay follows a structured reading philosophy. A paper is not only a sequence of pages, and it is not only a plain text document. It is a combination of layout, content, figures, tables, equations, references, and visual relationships. When reading a paper, users often depend on both the original layout and the extracted text. Removing either side makes the experience weaker.

For this reason, PaperQuay keeps the original PDF as the visual source of truth. At the same time, it builds a structured representation of the paper through parsing. The structured

representation can contain paragraphs, titles, lists, tables, captions, equations, and other block types. These blocks are connected back to the PDF through page indices and bounding boxes.

A simplified representation of a document block can be written as

$$B_i = (id_i, p_i, t_i, c_i, r_i), \quad (1)$$

where  $id_i$  is the block identifier,  $p_i$  is the page index,  $t_i$  is the block type,  $c_i$  is the block content, and  $r_i$  is the bounding box region on the PDF page. The bounding box can be represented as

$$r_i = (x_1, y_1, x_2, y_2). \quad (2)$$

This small piece of geometry is important. It allows PaperQuay to link a block of extracted content to the exact region of the original PDF page. As a result, linked reading can be based on document geometry instead of fragile text matching.

### 3 Library and Workspace Model

PaperQuay begins with the library. A first-time user is encouraged to open Settings and connect a local Zotero data folder. After the folder is connected, PaperQuay can display local library items, recent papers, collections, and available PDF attachments. Users can also add standalone PDFs when a paper is not managed by Zotero.

Each paper is treated as a workspace. A workspace may contain the original PDF path, parsed MinerU output, translation cache, AI summary, reading history, and other local state. This model is useful because academic reading is rarely completed in one session. A user may open a paper today, parse it, translate several sections, ask questions, and return to it a week later. PaperQuay is designed to make that return natural.

Layer	Purpose	Example
<b>Library</b>	Find and select papers	Zotero collection, recent papers, standalone PDFs
<b>Workspace</b>	Store paper-specific state	PDF path, parsed blocks, summary, translations
<b>Reader</b>	Perform active reading	PDF view, block view, assistant, translation
<b>Cache</b>	Reuse generated results	MinerU JSON, translated blocks, AI summaries

Table 1: The basic workspace model used by PaperQuay.

This structure allows PaperQuay to behave like a desktop research tool rather than a temporary browser page.

### 4 Structured Reading with MinerU

MinerU parsing is one of the core workflows in PaperQuay. When a PDF is parsed, the paper is converted into a collection of structured blocks. These blocks preserve their position on the PDF page, which makes them useful for linked navigation, block-level translation, and structured summarization.

A document can be described as a sequence of pages:

$$D = \{P_1, P_2, \dots, P_n\}. \quad (3)$$

Each page contains a sequence of blocks:

$$P_j = \{B_{j,1}, B_{j,2}, \dots, B_{j,m}\}. \quad (4)$$

The reader does not need to understand these formulas to use the application. However, this model explains why PaperQuay can connect the PDF and the extracted text reliably. The connection is not based primarily on searching for matching strings. It is based on the page index and bounding box of each block.

## 5 Linked PDF and Block Reading

The linked reading view is the central reading experience in PaperQuay. The left side displays the original PDF. The right side displays structured blocks extracted from the same paper. The two sides are connected by block identifiers and geometric positions.

A simple coordinate conversion is used when drawing PDF highlights. If a bounding box is defined in the original PDF coordinate system, and the page is rendered at a different size, PaperQuay maps the box into screen coordinates:

$$left = x_1 \cdot scale_x, \quad (5)$$

$$top = y_1 \cdot scale_y, \quad (6)$$

$$width = (x_2 - x_1) \cdot scale_x, \quad (7)$$

$$height = (y_2 - y_1) \cdot scale_y. \quad (8)$$

This conversion allows highlights to stay aligned with the rendered PDF page even when the page is scaled on screen.

## 6 Translation Workflows

PaperQuay supports two common translation workflows. The first is selection translation. When a user selects a small piece of text, the assistant can translate that selection without changing the rest of the reading context. This is useful when the reader only needs help with one sentence, one definition, or one difficult paragraph.

The second workflow is full-paper translation. After MinerU parsing is available, PaperQuay can translate the paper block by block. This approach is more structured than sending the whole paper as one long text. Each block keeps its relationship to the original document, and translated content can be displayed next to the corresponding source block.

A simplified translation mapping can be described as

$$T(B_i) = B'_i. \quad (9)$$

Here,  $B_i$  is the original block and  $B'_i$  is the translated block. Because both blocks share the same logical identity, PaperQuay can keep the translation aligned with the original paper structure.

## 7 AI Summary and Question Answering

AI assistance in PaperQuay is designed to support reading rather than replace reading. A summary can help the user form an initial mental model of the paper. A question assistant can help clarify a concept, explain a paragraph, compare sections, or identify the role of a method or result.

For example, a user might ask: “What problem is this paper trying to solve, and how is its method different from previous work?” After selecting a difficult paragraph, the user might ask the assistant to explain it in simpler language and list the key assumptions. These questions are most useful when connected to the current paper workspace.

## 8 Recommended Workflow for New Users

A new user should begin with the library rather than the reader. First, open Settings and connect the local Zotero folder. If Zotero is not available, add a standalone PDF. Then return to the Library and select one paper. It is better to start with one paper because the main features of PaperQuay are easiest to understand in a concrete reading session.

After opening a paper, check whether structured blocks are already available. If not, run MinerU Parse. Once blocks are available, switch to linked reading mode. Click a block on the right side and observe how the PDF moves to the corresponding location. Then click a PDF region and observe how the block view reacts. This interaction is the foundation of PaperQuay’s reading model.

1. Connect a library.
2. Select one paper.
3. Open the reader.
4. Run MinerU Parse.
5. Read with linked PDF and blocks.
6. Translate selected text when needed.
7. Generate an AI summary.
8. Use Q&A for difficult sections.
9. Return later and continue from the same workspace.

## 9 Limitations and Expectations

PaperQuay is a reading tool, not a guarantee that every paper will be parsed perfectly. PDF files vary widely in layout quality, embedded fonts, scanned images, mathematical notation, and table structure. MinerU parsing may work very well for some papers and less perfectly for others. AI summaries and translations may also contain mistakes, especially for highly technical or ambiguous content.

For serious academic work, the original PDF should remain the final reference. PaperQuay is designed to help the reader navigate, understand, translate, and organize the reading process. It should be used as an assistant to human judgment, not as a replacement for it.

## 10 Conclusion

PaperQuay combines a desktop paper library, a PDF reader, structured document parsing, linked navigation, translation, and AI-assisted reading into one environment. Its central idea is that academic reading benefits from structure. When a paper is represented as linked pages and blocks, the reader can move more easily between layout, text, translation, and explanation.

The best way to learn PaperQuay is to start with one paper. Connect a library, open the paper, parse it, and try the linked reading view. From there, translation, summaries, and Q&A become natural extensions of the same workspace.