Eli and Madi Dataset

Madeline Sullivan

October 2025

1 Data 1: Real-World PDF Document Collection

1.1 Overview

Data 1 is a custom-compiled dataset composed of real-world PDF documents collected from publicly available sources. The data set is designed to provide a broad representation of document layouts, structures, and content types for use in the development of optical character recognition (OCR) and text line recognition models.

The collection reflects the diversity and complexity typically encountered in practical document-processing scenarios, including variations in formatting, typography, image quality, and textual density.

1.2 Contents and Composition

The dataset includes a wide range of document types drawn from multiple domains, such as:

- Academic papers and reports
- Government and legal documents
- Business forms, invoices, and letters
- Technical manuals and specification sheets
- Books, brochures, and promotional materials

Each PDF file may contain one or more pages of text, tables, and graphics. The documents were intentionally selected to include a variety of languages, font styles, page layouts, and content complexity levels.

1.3 File Format and Structure

- Source format: PDF
- Derived formats: page-level and line-level images (PNG or JPEG)
- Annotation type: textual transcription for selected line images
- Average file size: 200 KB–3 MB per document
- Dataset size: customizable; currently several hundred documents

StreamingVLM: Real-Time Understanding for Infinite Video Streams

arXiv:2510.09608v1 [cs.CV] 10 Oct 2025

STREAMINGVLM: REAL-TIME UNDERSTANDING FOR INFINITE VIDEO STREAMS

Ruyi Xu[1*] **Guangxuan Xiao**[1*] **Yukang Chen**[2] **Liuning He**[1]
Kelly Peng[3] **Yao Lu**[2] **Song Han**[1,2]
[1]**MIT** [2]**NVIDIA** [3]**First Intelligence**
`https://github.com/mit-han-lab/streaming-vlm`

ABSTRACT

Vision-language models (VLMs) could power real-time assistants and autonomous agents, but they face a critical challenge: understanding near-infinite video streams without escalating latency and memory usage. Processing entire videos with full attention leads to quadratic computational costs and poor performance on long videos. Meanwhile, simple sliding window methods are also flawed, as they either break coherence or suffer from high latency due to redundant recomputation. In this paper, we introduce **StreamingVLM**, a model designed for real-time, stable understanding of infinite visual input. Our approach is a unified framework that aligns training with streaming inference. During inference, we maintain a compact KV cache by reusing states of attention sinks, a short window of recent vision tokens, and a long window of recent text tokens. This streaming ability is instilled via a simple supervised fine-tuning (SFT) strategy that applies full attention on short, overlapped video chunks, which effectively mimics the inference-time attention pattern without training on prohibitively long contexts. For evaluation, we build **Inf-Streams-Eval**, a new benchmark with videos averaging over two hours that requires dense, per-second alignment between frames and text. On Inf-Streams-Eval, **StreamingVLM** achieves a **66.18%** win rate against GPT-4O mini and maintains stable, real-time performance at up to 8 FPS on a single NVIDIA H100. Notably, our SFT strategy also enhances general VQA abilities without any VQA-specific fine-tuning, improving performance on LongVideoBench by +4.30 and OVOBench Realtime by +5.96.

1 INTRODUCTION

VLMs could power autonomous driving, embodied agents, and real-time assistants, but they face critical challenges: understanding near-infinite video, responding in real time stably. To accept infinite input, common ideas are Sliding Window Attention with or without overlapping. As shown in Figure 1: (a) *Full Attention* suffers from heavy memory and latency; (b) *Sliding Window (w/o Overlapping)* resets context frequently and breaks coherence; (c) *Sliding Window Attention (w/ Overlapping)* keeps recent tokens but recomputes attention many times, which hurts efficiency.

Aligning training with inference adds further challenges. Real streaming requires taking infinite visual input in real time and replying with very low delay, but training cannot use extremely long videos. Current approaches to KV cache eviction often lack alignment with the training phase. How to train on short videos and still enable the model to reason over very long streams remains underexplored. This leads to our core question: *How can we train VLMs to understand video chunks in real time and reason stably over infinite video, moving toward human-like intelligence?*

In this paper, we propose **StreamingVLM**, a unified framework that aligns training with streaming inference and a dataset curation pipeline. The key ideas are: (1) Train the VLM with full attention on short, overlapped video chunks. (2) At inference, use an attention sink and a sliding window with to handle infinite video, aligned with training. (3) Reuse past KV states and use contiguous position IDs to keep inference stable.

[*] Equal contribution

1

Figure 1: Dynamic Gaussian Fusion

1.4 Data Processing Workflow

Each PDF is converted into a set of image files using automated scripts. Text is extracted using OCR techniques (e.g., Tesseract or PyMuPDF) to create line-level input–output pairs for model training. The resulting data set therefore consists of:

- **Input:** Image of a text line (cropped from the PDF)
- **Output:** Corresponding text string

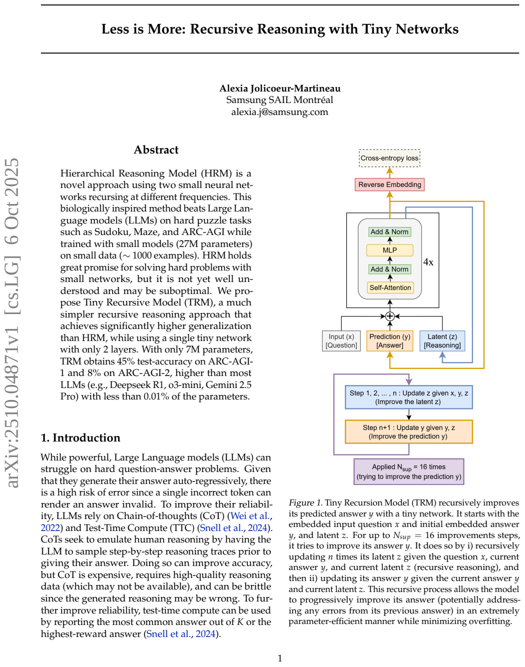
This preprocessing pipeline enables flexible training for both printed-text and mixed-format document recognition models.

1.5 Source and Licensing

All PDFs were obtained from openly accessible sources, such as:

- ()
- Internet Archive (`https://archive.org/details/texts`)
- STREAMINGVLM: REAL-TIME UNDERSTANDING ()
- Academic repositories such as arXiv ()

The documents were selected and used according to their open-access or public-domain licenses.



arXiv:2510.04871v1 [cs.LG] 6 Oct 2025

Less is More: Recursive Reasoning with Tiny Networks

Alexia Jolicoeur-Martineau
Samsung SAIL Montréal
alexia.j@samsung.com

Abstract

Hierarchical Reasoning Model (HRM) is a novel approach using two small neural networks recursing at different frequencies. This biologically inspired method beats Large Language models (LLMs) on hard puzzle tasks such as Sudoku, Maze, and ARC-AGI while trained with small models (27M parameters) on small data (∼ 1000 examples). HRM holds great promise for solving hard problems with small networks, but it is not yet well understood and may be suboptimal. We propose Tiny Recursive Model (TRM), a much simpler recursive reasoning approach that achieves significantly higher generalization than HRM, while using a single tiny network with only 2 layers. With only 7M parameters, TRM obtains 45% test-accuracy on ARC-AGI-1 and 8% on ARC-AGI-2, higher than most LLMs (e.g., Deepseek R1, o3-mini, Gemini 2.5 Pro) with less than 0.01% of the parameters.

1. Introduction

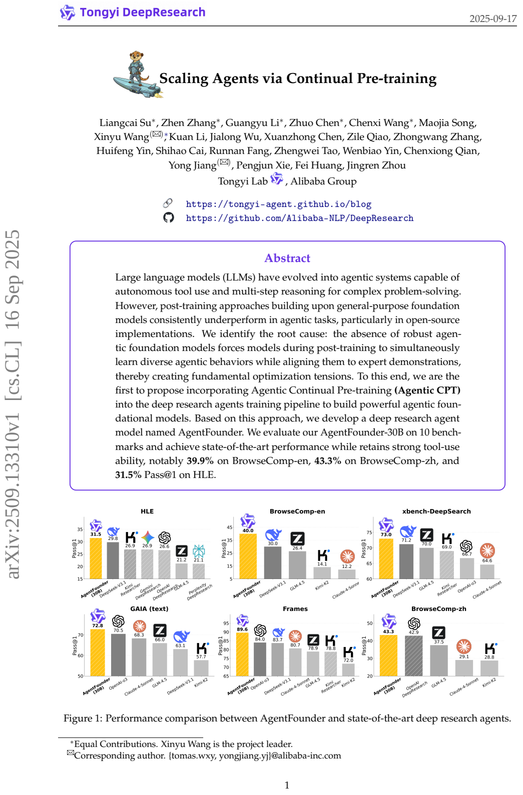
While powerful, Large Language models (LLMs) can struggle on hard question-answer problems. Given that they generate their answer auto-regressively, there is a high risk of error since a single incorrect token can render an answer invalid. To improve their reliability, LLMs rely on Chain-of-thoughts (CoT) (Wei et al., 2022) and Test-Time Compute (TTC) (Snell et al., 2024). CoTs seek to emulate human reasoning by having the LLM to sample step-by-step reasoning traces prior to giving their answer. Doing so can improve accuracy, but CoT is expensive, requires high-quality reasoning data (which may not be available), and can be brittle since the generated reasoning may be wrong. To further improve reliability, test-time compute can be used by reporting the most common answer out of K or the highest-reward answer (Snell et al., 2024).

Cross-entropy loss
Reverse Embedding
Add & Norm
MLP
Add & Norm
Self-Attention
4x
Input (x) [Question]
Prediction (y) [Answer]
Latent (z) [Reasoning]
Step 1, 2, ... , n : Update z given x, y, z (Improve the latent z)
Step n+1 : Update y given y, z (Improve the prediction y)
Applied N$_{sup}$ = 16 times (trying to improve the prediction y)

Figure 1. Tiny Recursion Model (TRM) recursively improves its predicted answer y with a tiny network. It starts with the embedded input question x and initial embedded answer y, and latent z. For up to $N_{sup} = 16$ improvements steps, it tries to improve its answer y. It does so by i) recursively updating n times its latent z given the question x, current answer y, and current latent z (recursive reasoning), and then ii) updating its answer y given the current answer y and current latent z. This recursive process allows the model to progressively improve its answer (potentially addressing any errors from its previous answer) in an extremely parameter-efficient manner while minimizing overfitting.

1

Figure 2: REAL-TIME UNDERSTANDING



Tongyi DeepResearch 2025-09-17

arXiv:2509.13310v1 [cs.CL] 16 Sep 2025

Scaling Agents via Continual Pre-training

Liangcai Su*, Zhen Zhang*, Guangyu Li*, Zhuo Chen*, Chenxi Wang*, Maojia Song,
Xinyu Wang(✉)*, Kuan Li, Jialong Wu, Xuanzhong Chen, Zile Qiao, Zhongwang Zhang,
Huifeng Yin, Shihao Cai, Runnan Fang, Zhengwei Tao, Wenbiao Yin, Chenxiong Qian,
Yong Jiang(✉), Pengjun Xie, Fei Huang, Jingren Zhou
Tongyi Lab, Alibaba Group

https://tongyi-agent.github.io/blog
https://github.com/Alibaba-NLP/DeepResearch

Abstract

Large language models (LLMs) have evolved into agentic systems capable of autonomous tool use and multi-step reasoning for complex problem-solving. However, post-training approaches building upon general-purpose foundation models consistently underperform in agentic tasks, particularly in open-source implementations. We identify the root cause: the absence of robust agentic foundation models forces models during post-training to simultaneously learn diverse agentic behaviors while aligning them to expert demonstrations, thereby creating fundamental optimization tensions. To this end, we are the first to propose incorporating Agentic Continual Pre-training **(Agentic CPT)** into the deep research agents training pipeline to build powerful agentic foundational models. Based on this approach, we develop a deep research agent model named AgentFounder. We evaluate our AgentFounder-30B on 10 benchmarks and achieve state-of-the-art performance while retains strong tool-use ability, notably **39.9%** on BrowseComp-en, **43.3%** on BrowseComp-zh, and **31.5%** Pass@1 on HLE.

Figure 1: Performance comparison between AgentFounder and state-of-the-art deep research agents.

*Equal Contributions. Xinyu Wang is the project leader.
✉Corresponding author. {tomas.wxy, yongjiang.yj}@alibaba-inc.com

1

Figure 3: Scaling Agents via

1.6 Use in This Project

This data set (**Data 1**) will serve as the basis for developing and evaluating a line-level text recognition model. Training By various types of real-world content aims to achieve robust generalization across document types, fonts, and layout structures.